



## Mathematical models and heuristic solutions for container positioning problems in port terminals

Kallehauge, Louise Sibbesen

*Publication date:*  
2008

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Kallehauge, L. S. (2008). *Mathematical models and heuristic solutions for container positioning problems in port terminals*. DTU Compute PHD

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Mathematical models and heuristic solutions for container positioning problems in port terminals

Louise K Sibbesen

Submitted in partial fulfillment of the requirements  
of the degree of Doctor of Philosophy

Technical University of Denmark  
Department of Management Engineering

February 2008



## Abstract

This PhD thesis is concerned with the container positioning problem (CPP) which consists in determining optimal sequences of positions and moves for containers in a single storage block of a terminal yard. The purpose of the thesis is to apply Operations Research (OR) methods for optimizing the CPP by constructing mathematical programming formulations of the problem and developing an efficient heuristic algorithm for its solution.

The thesis consists of an introduction, two main chapters concerning new mathematical formulations and a new heuristic for the CPP, technical issues, computational results, and conclusive remarks.

The introduction provides a basis for appreciating the presented work and sets out the scope, motivation, purpose, and contributions of the thesis. Furthermore, the CPP is defined and described, an overview of port container terminal issues in general is provided, and relevant literature concerning the subject is reviewed.

The research presented in this thesis is divided into two main parts: Construction and investigation of new mathematical programming formulations of the CPP and development and implementation of a new event-based heuristic for the problem.

The first part presents three mathematical programming formulations. First, a conceptual mixed integer linear programming (MIP) model for the entire port container terminal is presented. Subsequently, two models for the CPP are suggested: A MIP model and a binary integer linear programming (BIP) model. The models provide a basis for analyzing the CPP, demonstrating its complexity, and investigating potentials in model-based exact solution approaches. The models are solved by standard optimization software and the results as well as perspectives for alternative solution methods, making use of the models, are discussed.

The second part presents an efficient solution algorithm for the CPP. Based on a number of new concepts, an event-based construction heuristic is developed and its ability to solve real-life problem instances is established. The backbone of the algorithm is a list of events, corresponding to a sequence of operations in the storage block. This concept enables a representation of the time dimension of the problem which is very efficient. Furthermore, introducing a range of criteria for evaluating and selecting positions for containers makes both a highly effective and very flexible algorithm which is also robust to changes in parameters and input data. Two improvement routines are presented, one imbedded in the basic heuristic and the other constituting a repair algorithm with the purpose of improving an initial heuristic solution. The heuristic algorithm performance and a wide range of different planning strategies are investigated by solving a large number of test instances and real-life problems.

A total of 60 small-scale, 60 medium-scale, and 288 large-scale instances are introduced and used in the conduction of the computational experiments on the models and the heuristic algorithm.

Results from the model runs show that it is difficult to obtain optimal solutions to the CPP by solving the mathematical formulations using standard optimizers. Furthermore,

investigation of the potential of applying a relaxation approach indicates that this may not be a fruitful direction. Results from the heuristic runs proves the proposed algorithm very suitable for the CPP as good solutions are obtained within very short run times. Some important issues for further improvement of the heuristic algorithm are presented.

Conclusively it may be stated that the proposed mathematical models are complex and hard to solve by standard optimization software and that the presented heuristic algorithm is very robust and scalable and constitutes a highly efficient solution method for the CPP. The conclusive remarks are followed by some interesting perspectives for future research.

## Resume

Den danske titel på denne ph.d.-afhandling er “Matematiske modeller og heuristiske løsninger til containerpositioneringsproblemer i havneterminaler”. Afhandlingen omhandler containerpositioneringsproblemet (CPP), som består i at bestemme optimale sekvenser af positioner og flytninger for containere i et afgrænset område af en terminals lagerplads. Formålet med afhandlingen er at anvende operationsanalytiske metoder til at optimere CPP gennem konstruktion af matematiske programmeringsmodeller for problemet og udvikling af en effektiv heuristisk algoritme til at løse det.

Afhandlingen består af en introduktion, to centrale kapitler omhandlende nye matematiske formuleringer og en ny heuristik for CPP, redegørelse for en række tekniske aspekter, beregningsmæssige resultater samt en konklusion.

Introduktionen giver en indføring i emne, motivation og formål med det præsenterede arbejde samt en redegørelse for afhandlingens væsentligste bidrag. Desuden defineres og beskrives CPP, der gives et overblik over emner vedrørende containerterminaler, og relevant litteratur inden for området gennemgås.

Det præsenterede forskningsarbejde består af to centrale dele: konstruktion og udforskning af nye matematiske formuleringer for CPP samt udvikling og implementering af en ny eventbaseret heuristik for problemet.

I den første del tilvejebringes tre matematiske programmeringsmodeller. Først opstilles en konceptuel blandet heltalsmodel for hele containerterminalen. Derefter præsenteres to modeller for CPP: en blandet heltalsmodel og en binær heltalsmodel. Modellerne skaber en basis for at analysere CPP, at fastlå problemets kompleksitet og for at udføre en grundig undersøgelse af potentielle muligheder for modelbaserede eksakte løsningsmetoder. Modellerne løses med standard-optimeringssoftware, og såvel resultater som perspektiver for alternative løsningsmetoder diskuteres.

I den anden del præsenteres en effektiv løsningsalgoritme for CPP. En eventbaseret konstruktionsheuristik, som bygger på en række nye koncepter, udvikles, og dens egnethed til løsning af virkelige probleminstanser konstateres. Grundstenene i algoritmen udgøres af en eventliste, som styrer en given sekvens af operationer i lagerblokken. Dette koncept gør det muligt at repræsentere problemets tidsdimension på en meget effektiv måde. Desuden gør en række kriterier for evaluering og udvælgelse af containerpositioner metoden yderst effektiv og fleksibel samt robust i forhold til ændringer i parametre og data. To forbedringsrutiner præsenteres, den ene en integreret del af den grundlæggende heuristik og den anden en overbygningsalgoritme, der forsøger at forbedre en initial heuristisk løsning. Heuristikkens evne i forhold til løsningskvalitet og tidsmæssig effektivitet testes, og en række forskellige planlægningsstrategier undersøges gennem løsning af virkelige probleminstanser.

I alt 60 små, 60 mellemstore og 288 store probleminstanser introduceres og bruges i udførelsen af beregningsmæssige eksperimenter med modeller og den heuristiske algoritme.

Resultater fra modelkørslerne viser, at det er vanskeligt at opnå optimale CPP-løsninger ved at køre modellerne med standardløserne. Desuden indikerer undersøgelser af

potential for at anvende en relaxeringsmetode på problemet, at dette ikke umiddelbart er en lovende tilgang. Resultater fra de heuristiske kørsler fastslår, at den præsenterede algoritme er yderst velegnet til løsning af CPP, idet der opnås gode løsninger inden for meget korte køretider. Endelig belyses en række vigtige aspekter for videreudvikling af heuristikken.

Det konkluderes, at de fremsatte matematiske modeller er komplekse og svære at løse med standard-optimeringssoftware, samt at den præsenterede heuristiske algoritme er meget robust, ikke er følsom overfor problemstørrelse, og at den udgør en yderst effektiv løsningsmetode for CPP. Konklusionen følges op med nogle interessante perspektiver for fremtidig forskning inden for området.

## Acknowledgements

The research presented in this thesis has been carried out with inspiration and support from a number of people and institutions. The main part of the work has been carried out at the former Centre for Traffic and Transport and the thesis has been completed at the Department of Management Engineering at the Technical University of Denmark.

My time at Centre for Traffic and Transport has been both inspiring and cheerful thanks to the people I have had fruitful scientific discussions with and, not forgetting, due to the many good times after working hours.

I would like to express my deep thanks to the people from academia and the industry who have provided great inspiration and valuable input to my research. During my stay at Institut für Wirtschaftsinformatik at Universität Hamburg I gained important knowledge of present approaches to solving planning problems in port container terminals through my meetings with Professor Dr. Stefan Voß and his colleagues who also gave me the opportunity to visit Altenwerder Container Terminal in Hamburg where I got a valuable insight into the practical aspects and handling of real-life situations. Furthermore, I am grateful for the many discussions I have had with Finn Nørgaard from InPort and the industrial knowledge he and his colleagues has provided.

I would also like to express my thanks to the people at Zuse Institute Berlin for inspiration and fruitful discussions during my stay there. Particularly, my deep thanks go to Dr. Marc Pfetsch and Dr. Ralf Borndörfer who gave me important inspiration to my work on mathematical modeling and perspectives for exact solution approaches. I am also grateful for the valuable discussion I had with assistant professor Marco Lübbecke from Technische Universität Berlin who inspired me in the initial phase of developing the heuristic algorithm presented in the thesis.

I am very grateful for the PhD scholarship provided by the Technical University of Denmark and for the support by the departments where my research has been performed.

February 20, 2008

Louise K Sibbesen





# Contents

|          |                                                                     |           |
|----------|---------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                 | <b>13</b> |
| 1.1      | The container positioning problem . . . . .                         | 16        |
| 1.2      | Port container terminals . . . . .                                  | 18        |
| 1.3      | Literature review . . . . .                                         | 25        |
| 1.4      | Outline of the thesis . . . . .                                     | 30        |
| <b>2</b> | <b>New mathematical formulations of container terminal problems</b> | <b>33</b> |
| 2.1      | A MIP model for port container terminals . . . . .                  | 35        |
| 2.1.1    | The CTP model . . . . .                                             | 38        |
| 2.1.2    | Model analysis . . . . .                                            | 42        |
| 2.2      | A MIP model for the CPP . . . . .                                   | 43        |
| 2.2.1    | The CPP model . . . . .                                             | 45        |
| 2.2.2    | Model analysis . . . . .                                            | 49        |
| 2.3      | A time-discretized BIP model for the CPP . . . . .                  | 50        |
| 2.3.1    | The CPPT model . . . . .                                            | 52        |
| 2.3.2    | Model analysis . . . . .                                            | 55        |
| 2.4      | Conclusive remarks . . . . .                                        | 57        |
| <b>3</b> | <b>A new event-based heuristic for the CPP</b>                      | <b>61</b> |
| 3.1      | Concepts . . . . .                                                  | 64        |
| 3.1.1    | System time . . . . .                                               | 64        |
| 3.1.2    | List of events . . . . .                                            | 64        |
| 3.1.3    | Evaluation of candidate positions . . . . .                         | 68        |
| 3.1.4    | Selection of the best possible position . . . . .                   | 72        |
| 3.2      | The algorithm . . . . .                                             | 74        |
| 3.2.1    | Input data and configuration . . . . .                              | 75        |
| 3.2.2    | Pseudo code . . . . .                                               | 78        |
| 3.3      | Improvements . . . . .                                              | 86        |
| 3.3.1    | Pesudo code for procedures to avoid rejections . . . . .            | 90        |
| 3.3.2    | Pseudo code for shifting routine . . . . .                          | 91        |
| 3.4      | Conclusive remarks . . . . .                                        | 98        |

|          |                                                                |            |
|----------|----------------------------------------------------------------|------------|
| <b>4</b> | <b>Computational platforms and test problems</b>               | <b>99</b>  |
| 4.1      | Computational platforms . . . . .                              | 99         |
| 4.2      | Data and test problems . . . . .                               | 100        |
| 4.2.1    | Problem types . . . . .                                        | 100        |
| 4.2.2    | Problem classes . . . . .                                      | 100        |
| 4.2.3    | Naming conventions . . . . .                                   | 105        |
| <b>5</b> | <b>Computational results</b>                                   | <b>107</b> |
| 5.1      | Results for the CPP and CPPT models . . . . .                  | 108        |
| 5.1.1    | Results for the CPP model . . . . .                            | 109        |
| 5.1.2    | Results for the CPPT model . . . . .                           | 116        |
| 5.1.3    | Results for the models with extended run time limit . . . . .  | 124        |
| 5.1.4    | Results for the models with relaxed life constraints . . . . . | 125        |
| 5.1.5    | Conclusive remarks on the model results . . . . .              | 128        |
| 5.2      | Results for the heuristic algorithm . . . . .                  | 131        |
| 5.2.1    | Tuning phase . . . . .                                         | 132        |
| 5.2.2    | Small- and medium-scale class results . . . . .                | 133        |
| 5.2.3    | Large-scale class results . . . . .                            | 151        |
| 5.2.4    | Conclusive remarks on the heuristic results . . . . .          | 202        |
| 5.3      | Comparison and conclusion . . . . .                            | 203        |
| 5.3.1    | Conclusive remarks . . . . .                                   | 206        |
| <b>6</b> | <b>Conclusions</b>                                             | <b>209</b> |
| 6.1      | Perspectives . . . . .                                         | 210        |
|          | <b>Bibliography</b>                                            | <b>213</b> |
| <b>A</b> | <b>Overview of OR developments</b>                             | <b>219</b> |
| A.1      | Background on OR . . . . .                                     | 219        |
| A.2      | Heuristics and metaheuristics . . . . .                        | 225        |
| <b>B</b> | <b>Input data</b>                                              | <b>227</b> |
| <b>C</b> | <b>CTP model solution</b>                                      | <b>229</b> |
| <b>D</b> | <b>CPP model solutions</b>                                     | <b>231</b> |
| D.1      | Small-scale class instances . . . . .                          | 231        |
| D.2      | Medium-scale class instances . . . . .                         | 259        |
| <b>E</b> | <b>CPPT model solutions</b>                                    | <b>289</b> |
| E.1      | Small-scale class instances . . . . .                          | 289        |
| E.2      | Medium-scale class instances . . . . .                         | 304        |

|          |                                                                           |            |
|----------|---------------------------------------------------------------------------|------------|
| <b>F</b> | <b>CPP and CPPT model solutions with extended time limit</b>              | <b>323</b> |
| F.1      | Scaled 10 containers subclass instances solved by the CPP model . . . . . | 323        |
| F.2      | Tight 10 containers subclass instances solved by the CPPT model . . . . . | 324        |
| <b>G</b> | <b>CPP and CPPT model solutions with relaxed lifo constraints</b>         | <b>327</b> |
| G.1      | Medium-scale class instances solved by the CPP model . . . . .            | 327        |
| G.2      | Medium-scale class instances solved by the CPPT model . . . . .           | 342        |
| <b>H</b> | <b>Heuristic tuning results</b>                                           | <b>353</b> |
| H.1      | Small-scale class instances . . . . .                                     | 353        |
| H.2      | Medium-scale class instances . . . . .                                    | 368        |
| <b>I</b> | <b>Heuristic solutions</b>                                                | <b>383</b> |
| I.1      | Small-scale class instances . . . . .                                     | 383        |
| I.2      | Medium-scale class instances . . . . .                                    | 396        |
| I.3      | Large-scale class instances . . . . .                                     | 413        |



# Chapter 1

## Introduction

The subject of this thesis is the application of Operations Research (OR) methods for optimizing the handling of containers in a port container terminal yard. In particular, the thesis considers the problem of determining optimal sequences of positions for containers in a storage block by minimizing the total cost of positioning and moving the containers within the block. This problem is referred to as the container positioning problem (CPP). The objective of the CPP is to minimize the total transportation time and the number of moves between positions, the latter being referred to as reshuffles.

After arriving at the terminal, each container is assigned to a certain storage block where it is placed at a position and possibly reshuffled a number of times before being removed from the block and loaded onto a vehicle or a vessel for further transportation. The terminal yard is often divided into several blocks. In this thesis, the CPP is defined with respect to a single storage block.

An important characteristic of the CPP is the last-in first-out (lifo) principle, requiring that the last placed container in a stack must be removed from the position before container(s) below it can be moved. This aspect implies some significant challenges and makes the CPP a highly complex problem both from a mathematical modeling point of view and from a practical point of view.

Today, manual planning based on experience and rules of thumb rather than optimization methods is used in most - if not all - port container terminals. However, the heavy growth in containerized transportation witnessed through the past decades has led to an increased need for efficient planning methods to assist in handling containers effectively and to strengthen the competitiveness of the port. The CPP constitutes a bottleneck and presents the terminal managers with a very complex problem. Therefore, realizing that present planning methods do not suffice in achieving efficient storage and reshuffle plans for containers in the yard, this thesis argues that it is highly relevant to investigate mathematical optimization methods for the CPP.

The motivation of this thesis is the industrial relevance of developing advanced planning tools that can optimize the planning process in port container terminals. The purpose of the current thesis is to contribute to the research concerning container positioning problems and yard optimization by presenting mathematical models which enable

both analysis of the problems and investigating potential approaches to their solution, and to propose efficient solutions methods of relevance to practical application.

The thesis consists of two main parts: Construction and test of new mathematical programming formulations of the CPP and development and implementation of an efficient heuristic algorithm for the problem. The following sections provide an overview of the two parts of the thesis and present their main contributions.

## **Part 1: Construction and test of mathematical programming formulations**

When facing a problem not yet subject to standardized mathematical formulations and solution techniques, it is valuable to construct such formulations to provide a basis for better understanding, analysis, and further investigation. Clearly, a lot of time can be spent - and possibly wasted - on developing advanced solution procedures if the problem turns out to be well-structured and easy to solve by standard routines. Therefore, the first approach to a no yet well-studied problem should be to construct mathematical models for it and investigate whether standard solvers using branch and cut algorithms are suitable for their solution, possibly by testing different approaches to solving the compact algebraic formulations.

In this thesis a compact model is defined as a mathematical programming formulation of polynomial size, i.e. with a polynomial number of variables and constraints. In contrast to such a formulation are models with exponentially many variables or constraints, e.g. the formulation of the traveling salesman problem (TSP) with subtour elimination constraints defined on arc variables [62].

The modeling phase of the study consists in developing mathematical optimization models for the CPP, based on existing research concerning related problems as well as the knowledge of practical problem issues which is gained by visiting port container terminals and communicating with people from the industry and academia. A complete overview of the modeling process is presented in chapter 2.

First, a preliminary broad modeling approach to the entire port container terminal provides an important conceptual basis for understanding some problem structures and identifying complex aspects of the problem. The model is a mixed integer linear programming (MIP) model and is described in section 2.1. Next, narrowing down to the scope of the CPP, two different modeling approaches, representing the problem structure - especially the time dimension - in different ways, are constructed. The first CPP model is a MIP model and the second is a binary integer linear programming (BIP) model. The models are described in sections 2.2 and 2.3.

The first model is not subject to further investigation as it does not capture the central problem of the thesis. The two models for the CPP, on the other hand, are tested using standard optimization software and potential exact solution approaches, alternative to solving the compact models in their entirety, are investigated. The solution process and results from the computational study of the models are reported in section 5.1.

Main contributions from part 1 are the following. Two new mathematical programming formulations of the CPP are introduced and compared. The lifo principle and the

time dimension are identified as major obstacles in the effort of creating efficient compact models for the CPP. The problem is analyzed based on the models and its complexity is established. A large number of benchmark problem instances of the CPP are designed and constructed, representing the different arrival and departure time patterns that can occur in a given instance of the problem. The models are implemented in the standard algebraic modeling language Xpress-Mosel, their correctness is ensured by thorough investigation of 60 small-scale instance solutions, and their performance is evaluated by solving 60 medium-scale problem instances. The computational efficiency of the models when solved by the standard solver Xpress-optimizer is evaluated and the potential of applying a solution approach, different from the standard compact model optimization, is investigated by relaxation of a set of complicating constraints. Furthermore, part 1 of the PhD study forms the basis for understanding the CPP, realizing the complexity of the problem, and developing alternative solution approaches which leads us to the second part of the thesis: The development of an efficient event-based heuristic algorithm for the CPP.

## **Part 2: Development and implementation of an efficient heuristic algorithm**

As an alternative to solving compact mathematical models for a problem, tailored solution methods may be developed. In general, two different directions can be investigated: Exact and heuristic algorithms.

For some problems very efficient exact solution algorithms can be developed, e.g. decomposition methods making use of simpler subproblem structures that provide a good approximation of the original problem. However, if it is not possible to identify good relaxations or decompositions of the problem, it may be difficult to achieve optimal results in reasonable computation time. In general, the advantage of heuristic algorithms is their ability to solve a problem very fast - in some cases even being able to solve it. The quality of a heuristic solution clearly depends partly on the algorithm design and implementation and partly on the problem structure as some problems seem to interact better with exact approaches than heuristics and vice versa.

Considering the importance of short computation times when solving real-life problems and implementing results in the industry, and with an application approach being part of the motivation for this PhD study, a heuristic algorithm for the CPP and an improvement routine are developed. Furthermore, certain indications from the model analyses turn out to point in the direction of heuristic solution methods being quite suited for the specific problem investigated. Chapter 3 presents the heuristic algorithm.

The algorithm is implemented in a standard programming language and a thorough investigation of its capacity and performance as well as various possibilities for improving procedures and subroutines is conducted. Computational results for the heuristic algorithm are presented in section 5.2.

Main contributions from part 2 are the following. Based on the knowledge gained from analyzing the proposed models, new concepts for handling the operations in a CPP instance are introduced. An event-based approach enables getting around the complexity of the time dimension and ensures handling of the given sequence of operations in an



efficient and structured way. A number of carefully selected decision criteria leading to a minimal number of reshuffles are proposed, and a set of penalty terms that can be adjusted to reach a desired outcome is presented. A range of tuning parameters makes it possible to apply a wide range of strategies that may suit the decision maker. These concepts result in a highly robust heuristic algorithm that is able to solve real-life problem instances in very short computation time. Furthermore, as opposed to solving the compact models for the CPP, the heuristic algorithm is very scalable in terms of problem size which makes it highly relevant for industrial application. Two improvement techniques are developed, each addressing an important issue when solving the CPP: Avoiding rejection of containers if feasibility requirements are not fulfilled and improving on-time performance of a given heuristic solution. The performance of the heuristic algorithm and the improvement routines is tested by solving a wide range of large-scale real-life problem instances and a number of different planning strategies are investigated. Furthermore, the heuristic algorithm is compared to the exact solutions of the models, demonstrating the approach as very well-suited for the CPP.

### **Structure of the remaining chapter**

The following section presents the CPP addressed in this thesis. Subsequently, section 1.2 concerns the context of the CPP: Port container terminals, a background knowledge of containers and the shipping industry in general, and a brief introduction to planning problems and practice in port container terminals. Section 1.3 provides an overview of relevant literature related to the CPP as well as survey papers concerning port container terminal issues. Finally, section 1.4 outlines the remaining chapters of this thesis. Furthermore, for managers of port container terminals and other practitioners who may not be familiar with OR methods, appendix A provides some useful knowledge of developments within the OR field.

## **1.1 The container positioning problem**

The CPP consists in determining optimal positions and move sequences for containers in a single storage block of the terminal yard. The objective of realizing the lowest possible costs of positioning and reshuffling containers is achieved by minimizing the effort of handling the containers, i.e. minimizing the number of reshuffles and/or the time spent on moving the containers.

The restrictions, constituting the problem, represent physical as well as logical requirements. Containers must be picked up and dropped of at the right times, corresponding to the arrival and departure times of their modes of transportation to and from the block. Each container must be placed at a position and stored there until possibly moved to another position. At each position, a number of containers can be stacked on top of each other, the maximum level depending on the stacking equipment. When retrieving a container that is buried below one or several other containers at a position, the top containers must be removed first. This lifo principle is illustrated in figure 1.1 for two

containers,  $c$  and  $c'$ , at the same position. Only four situations are allowed, two of them, (A) and (B) are depicted in the figure. In these two cases, container  $c$  is placed at the position first, in (A) implying no time overlap with the storage time for container  $c'$ , and in (B) giving rise to the lifo restrictions. Analogous to situations (A) and (B) are cases where container  $c'$  is placed before  $c$  at the position. Only overlap between containers' storage times (situation (B)) imply the lifo restrictions, ensuring that a container placed on top of another one is removed before the lower container.

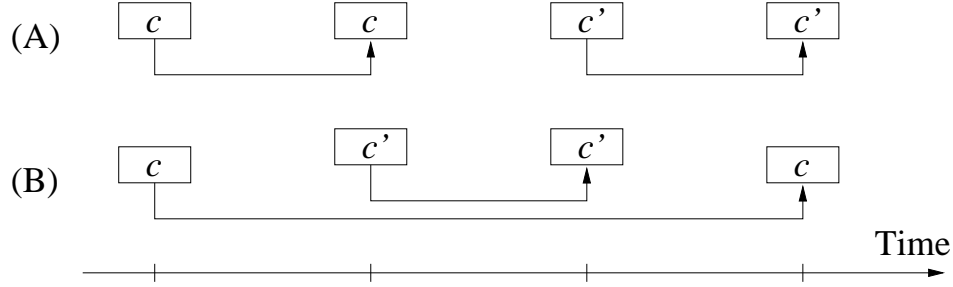


Figure 1.1: The lifo principle for a pair of containers  $c$  and  $c'$ . If both containers are placed at position  $p$  at some point in time, only four situations, (A) and (B) where  $c$  is placed first, and equivalent cases where  $c'$  is placed first, are allowed. In (A) no time overlap necessitates lifo restrictions, and in (B) the overlap between the containers' storage times at the position implies the lifo restrictions, ensuring that the last positioned container is removed from the stack before the lower one.

The structure of the CPP, addressed in this thesis, can be characterized as follows and illustrated in figure 1.2. A storage block consists of a number of positions in the horizontal plane, denoted  $i = 0, \dots, I$  with  $i = 0$  corresponding to the arrival place in one end of the block,  $i = I$  corresponding to the departure place in the other end. Actual positions in the block are  $i = 1, \dots, I - 1$  where the last position is denoted  $P = I - 1$ .

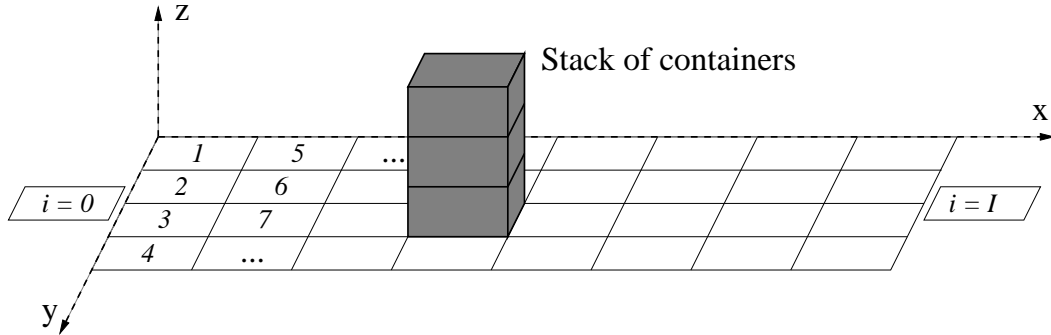


Figure 1.2: Principal structure of a storage block consisting of a number of positions ( $i = 0, \dots, I$ ) in the horizontal plane. At each actual position ( $1, 2, \dots, P$ ), a number of containers can be stacked.  $i = 0$  and  $i = I$  correspond to the arrival and departure place respectively.

Solutions to the CPP define sequences of positions and moving times for each container. The first move in such a sequence is the move from the arrival place  $i = 0$  to the first position for the container,  $p_1 \in \{1, 2, \dots, P\}$ . Subsequently, a number of reshuffles to other positions may be performed. The last move in the sequence is destined for the departure place  $i = I$ . Let  $p_n$ ,  $n \geq 1$  denote the last position for the container in the block. Clearly, the moves  $0 \rightarrow p_1$  and  $p_n \rightarrow I$  are unavoidable as each container must be placed in the storage block at least once. The goal is, therefore, to minimize the intermediate moves between block positions, i.e. the number of reshuffles, equivalent to minimizing the number of positions assigned to each container.

Given a sequence of moves, the corresponding sequence of moving times is as follows. The first move starts at the arrival time for the container and has a duration corresponding to the required time to move the Manhattan distance from  $i = 0$  to  $p_1$ . The second move starts after a certain storage time at position  $p_1$  and ends after the time interval required to move to the next position  $p_2$  or to the departure place  $I$  if the container is not reshuffled. The final move is required to end at  $I$  at the departure time for the container and, therefore, starts from the last position  $p_n$  the respective transportation time interval in advance.

After the above description of the CPP, the following section provides an overview of the context in which the problem occurs: Port container terminals.

## 1.2 Port container terminals

As a result of globalization and economic growth, the worldwide freight transportation has increased markedly through history. The introduction of containers has enabled standardized transportation and lightened terminal operations considerably which has led to further increases in shipped quantities.

The shipping industry has a natural monopoly on intercontinental heavy freight transportation and in line with the continuously growing demand for containerized transportation, new port container terminals are still being established to serve customers all over the world. Due to the heavy increase in seaborne transportation, the throughput in port container terminals has exploded during the past few decades. Figure 1.3 provides an overview of ten years' growth in port container terminal throughput in the ten largest ports worldwide. Especially Chinese ports have seen a tremendous increase in freight throughput. Furthermore, figure 1.4 provides an overview of the progress of international seaborne trade from 1970 to 2005 and figure 1.5 shows the annual increase in the last four years.

Containerization is, without doubt, among the most important contributions to intermodal freight transportation, enabling cheaper and faster service due to reduced cargo handling and damages - especially at transshipment points - and improved security. The below section takes the reader through the history of container developments and the following section provides an overview of operations and planning problems occurring in port container terminals.

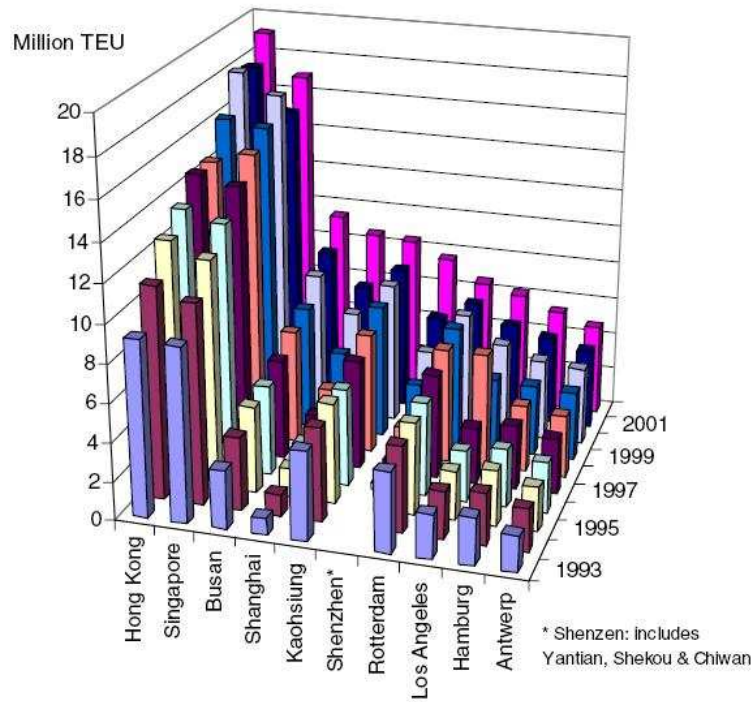


Figure 1.3: Container turnover in the ten largest port terminals in the world from 1993 to 2002, figure from [53].

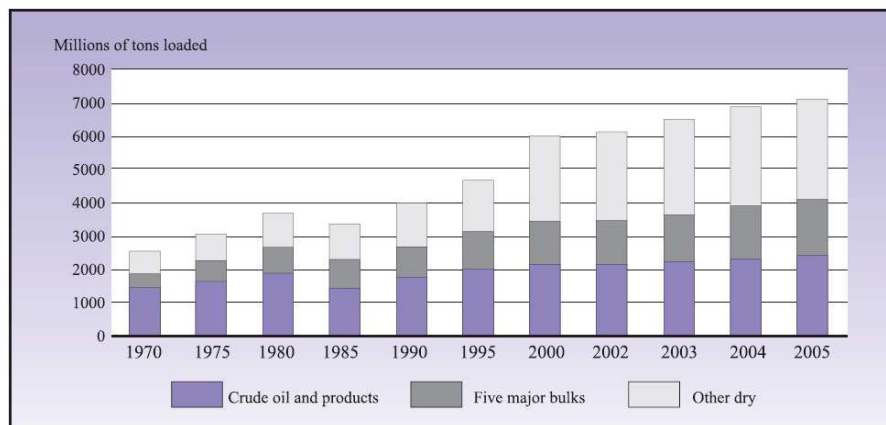


Figure 1.4: International seaborne trade for selected years, 1970 - 2005, figure from [8].

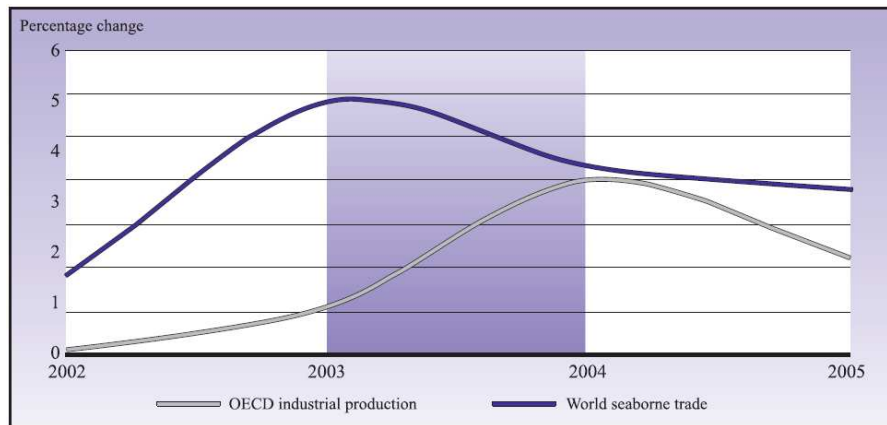


Figure 1.5: Annual change in OECD industrial production and world seaborne trade, 2002 - 2005, figure from [8].

## Background on container shipping

The invention of the container was a response to the need for a low cost transportation system, minimizing freight handling, transshipments, paperwork, and inspections, protecting perishable and fragile commodities, reducing breakage, waste, and theft, and enabling transportation of all kinds of goods - large or small quantities, heavy or light, in packs or in bulk. The container, fulfilling all these requirements, has been a permanent element in intermodal freight transportation since it entered the business half a century ago [56].

Intermodal freight transportation is shipping commodity by use of several transportation modes: Rail, ship, and/or truck. Furthermore, the term refers to integrated transportation systems where goods are not handled at mode changes, which is why the container has led to considerably higher efficiency in the industry [61].

Prior to the advent of containers, cargo was transported in bulk and - after the introduction of pallets - in different sorts of standardized packages. This made transportation, and especially transshipment, difficult and time-consuming. In the shipping industry, introduction of containers led to a dramatic reduction in ship's turn-round times (i.e. unloading and loading in ports plus potential waiting) from several days to only a few hours. Likewise, in the railway and trucking industry, use of containers led to faster service, less damage, waste, pilferage, and lower costs, to name a few advantages [57].

Pioneers in the early days of containers were Malcom McLean, a road transportation manager from North Carolina, and the White Pass & Yukon Rail Route (WP&YR), a railroad and river boat company, operating the isolated rail route between Skagway in Alaska and Whitehorse in the Canadian Yukon territory, established immediately after the Klondike gold rush at the end of the 19th century. Malcom McLean got the idea of stacking trailers on trains for long distance railroad transportation and widened the

concept by converting two small ships for trailer transportation. On August 31 1958, his first container vessel sailed from San Francisco to Honolulu [57, 61].

Contemporary with the initiatives of Malcom McLean, in 1955, WP&YR built and tested the first containers on the railway between Whitehorse and Skagway which proved to be a great success as the cargo arrived undamaged to its destination. Later that year, the company extended the transportation system by purchasing a container vessel and, thereby, creating the first containerized freight transportation link and initiating the world's first intermodal transportation system [56, 61].

The first containers measured only 6' x 8' x 7' so perceptible changes have characterized the developments of containers as we know them today. In the 1960's, the standard container size of 20' x 8' x 8'6" was agreed on and today, practically all container types are measured in twenty feet equivalent units (TEUs). Figure 1.6 shows a standard 20' (1 TEU) dry container. The standardization of freight containers belong under the International Organization for Standardization (ISO), established in 1946 by delegates from 25 countries with the purpose "to facilitate the international coordination and unification of industrial standards", officially beginning to work in February 1947 [25, 57, 56].



Figure 1.6: 20' (1 TEU) dry container.

In line with the heavy increase in containerized freight transportation, development of larger container ships, and establishment of port terminals throughout the world, the International Maritime Organization (IMO), formally founded at an international conference in Geneva in 1948 with the object of studying maritime safety - later also pollution issues by developing international regulations - has undertaken several studies of containerized freight transportation at sea with the container itself as the main focus [24, 57].

Throughout the 20th century the shipping industry has seen steady growth and, especially in the last decades, a tendency towards consolidation. The container pioneer Malcom McLean established the first container shipping company Sea-Land in the early 1960's and started operating vessel fleets of ships with a capacity of 2,000 TEU each in 1972. Today, the world's largest shipping company is Maersk Line, founded by recently retired chairman Mærsk Mc-Kinney Møller's father Arnold Peter Møller and grandfather captain Peter Mærsk Møller in 1904. The company's present name is the result of the overtaking of Royal P&O Nedlloyd N.V., the last of a long series of acquisitions, including

the buying of the Malcom McLean company Sea-Land Service Inc. in 1999 [44, 57].

Today the containerized freight transportation is globally standardized to a large extent on account of the container. The system of ISO standard containers enables smooth intermodal transportation and greatly reduces freight handling. The vast majority of containers used today measure 20' x 8' x 8'6" and 40' x 8' x 8'6". Also large "high cube" container containers measuring 40' x 8' x 9'6" and 45' x 8' x 9'6" have entered the industry in order to increase profit. In addition, different types of containers such as dry and reefer containers are used for various purposes [44].

### Operations and planning problems in container terminals

In general, three types of operations are carried out in port container terminals: Quay-side, yard, and landside operations. The quayside of the port terminal covers the quay, buffer zones for quick ship service in some ports, and areas for transportation between quays and storage facilities. The yard is typically divided into a number of storage blocks where containers of different sizes and types are stored. The landside covers the area between the storage blocks and the gates and railway zone. Figure 1.7 provides an overview of a principal port layout, characteristic for an automated terminal with the main part of the equipment being operated automatically rather than manually.

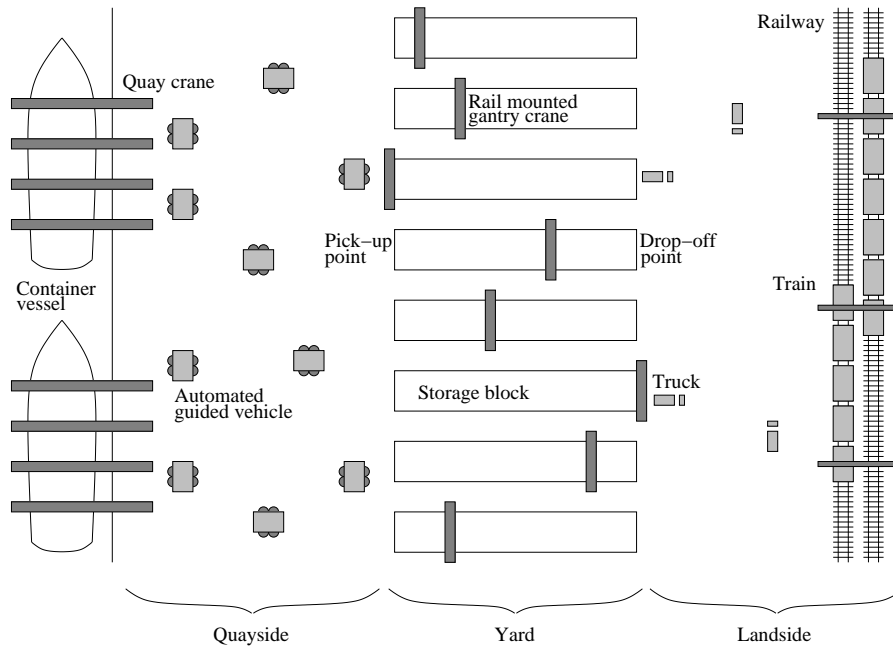


Figure 1.7: Principal layout of an automated port container terminal (not true to size), figure inspired by [53]. Layout and equipment differ from port to port.

For inbound containers, typically import containers, the following operations and equipment types can be identified (the sequence of operations is reversed for outbound

containers, typically export containers): After a vessel is assigned to a berth, quay cranes (QCs) unload the containers and automated guided vehicles (AGVs) transport each container to one of the storage blocks' arrival place. Block cranes such as rail mounted gantry cranes (RMGs) move the containers from the arrival place and store them at some predetermined positions. Each container may be reshuffled a number of times before being moved to the departure place. From this final position, containers are loaded onto trucks or chassis by which they are transported to their destinations either by road or by rail [53].

### **Background on practice in port container terminals**

This section provides some practical insight into the aspects of port container terminals, exemplified by two cases: Altenwerder Container Terminal (ACT) in Hamburg, a large automated terminal, and Oslo Container Terminal (OCT), a medium-sized container terminal. The information has been collected through meetings and conversations with people from Hamburg University and InPort [60, 48].

The quay line of ACT in Hamburg is about 1.6 km long with room for four large vessels, each with a capacity of more than 8,000 TEU, and several smaller vessels with a capacity of about 200 TEU. Every week, about 5 large vessels, following a certain schedule, and 50 - 60 smaller vessels call at the port. At the vessels, containers are stored both below and above deck, all oriented in the same direction. Containers are typically not reshuffled aboard a vessel.

About 14 QCs are located at the quayside of the port. They are operated manually as experiments with full automation proved manual operation to be faster. QCs are clustered up to four cranes together and service one vessel at a time, each crane allocated to a certain quay lane to/from which the AGVs pick up or drop off containers.

Over 40 AGVs, each with a capacity of 60 tons, operate at the quayside of ACT, transporting containers between the quay and the storage blocks. The optimal AGV velocity has been determined to 4 m/s as higher speeds will require more operating space for each vehicle due to safety issues. An AGV has a capacity of 40 TEU and can, therefore, carry either one 40' or two 20' containers at a time [60].

The yard of ACT is divided into 22 storage blocks, each with 10 rows, 37 bays, and a stacking height of 4, resulting in a block capacity of over 1,400 TEU and a total port capacity of over 30,000 TEU. [60]. The OCT yard consists of two storage blocks, each with 9 rows, 75 bays, and a stacking height of 4, resulting in a total storage capacity of 5,400 TEU [48].

Certain areas may be reserved for special types of containers. In ACT, three blocks have electrical plug-ins for reefer containers, three bays at the landside end of each block are reserved for dangerous goods so that fire engines can reach them quickly in case of accidents, and special areas in the yard are used for empty containers and oversized freight that must be handled manually [60]. OCT applies a more flexible system where specific area allocations change from day to day and certain positions or bays may be reserved to containers for which there are no other positions at the time of a given operation [48].



Containers are stored in the port terminal for 4 - 5 days on average. Positionings of the containers are often determined by the yard manager or by use of an information software tool which does not include any optimization modules. In some ports, the strategy is to place outbound containers close to the quay and inbound containers close to the landside gate whereas other terminals prioritize fast positioning in order to continue servicing e.g. a vessel rather than making good strategic decisions during unloading. Containers have to be positioned with the opening end facing the waterside so that loading onto trucks does not prevent checking the content from the back of the vehicle.

Two RMGs, unable to cross each other, operate at each block in OTC whereas the RMGs in ATC blocks are of different height and, therefore, both can operate in the entire block independently of each other. RMGs are operated automatically at the waterside as AGVs connect to the block at exact points whereas unloading/loading of containers from/onto trucks or chassis on at the landside end of the block is performed manually by use of cameras due to the manually driven vehicles. The advantage of a double RMG system, applied in both ATC and OTC, is the effectivity of two cranes working together, handling more workload than a single crane is able to, and moreover, that during maintenance of one of the cranes, the other can still operate in the block. However, from a mathematical optimization perspective, this system makes a very complex problem - especially if the cranes are able to cross each other. Alternatives to the RMG systems are rubbertired gantry crane (RTG) and the straddle carrier (SC) systems where the manually operated cranes can access almost all parts of the terminal. The average block crane velocity varies between 10 km/h for RMGs and RTGs to about 50 km/h for SCs [60, 48].

At the landside of the terminal, trucks arrive and depart through the gate, also servicing trains if the terminal has railway facilities. Loading or unloading of trains is performed in the same way as when servicing vessels. As well as each container must be placed at a certain position in a vessel, there are strict requirements for placing containers at certain waggons on a train [48].

Port authorities can begin planning the container operations when arrival information for ships, trucks and trains is available. Schedules for large container vessels are known about one year in advance of arrival time in the port whereas schedules for smaller vessels are not available that well in advance. In any case, the port authorities are provided with information about ships' schedules and stowage plans months or weeks in advance and, therefore, planning unloading sequences and positioning of containers can begin in good time before the ships call at the port. However, changes may occur so that these plans have to be changed shortly before a ship's arrival or during the unloading of the vessel. The arrival times for trucks are somewhat more uncertain. Port authorities in large ocean terminals often demand information about arrival times 24 hours in advance whereas this is not required in many smaller feeder terminals where trucks arrive at the gate with very short notice. Trains usually follow a fixed schedule which makes arrival and departure times reliable [53].

However, there is a significant difference between practice in Asian and European/American port terminals. Asian port authorities usually do not allow deviations from the

scheduled time slots, allocated to vessels, trucks, and trains and, thus, data for such ports is generally very accurate and reliable. On the other hand, there is a flexibility in European and American port terminals which makes arrival and departure times for connecting transportation modes less restricted and, therefore, plans and schedules highly uncertain and unreliable [60].

Following these practical aspects, the below section provides an overview of relevant literature concerned with port container terminals and the CPP in particular.

### 1.3 Literature review

This section provides an overview of literature concerning optimization of port container terminal problems, the CPP in particular. In general, literature about descriptive methods, especially simulation, dominates the field. As this PhD study concerns normative methods for optimizing container handling in the terminal yard, the following literature review focuses on exact optimization and heuristic approaches to the CPP as well as surveys papers and broader oriented literature, dealing partly with the CPP.

Figure 1.8 provides a schematic overview of the type of literature relevant to the present work. The hatched areas correspond to normative methods and the relevant problem scopes are highlighted with boxes.

|       |          | METHODOLOGICAL APPROACH |            |             |            |
|-------|----------|-------------------------|------------|-------------|------------|
|       |          | Normative               |            | Descriptive |            |
|       |          | Optimization            | Heuristics | Simulation  | IT systems |
| SCOPE | Quayside |                         |            |             |            |
|       | Yard     |                         |            |             |            |
|       | Landside |                         |            |             |            |
|       | Reviews  |                         |            |             |            |

Figure 1.8: Schematic overview of literature relevant to this work. The hatched areas correspond to normative methods. Scopes dealt with in this literature review are highlighted with boxes.

There are few articles before 1990 dealing with normative methods while several papers about simulation or control systems can be found. The vast majority of the literature within the subject is from the last decade which is believed to be due to the significant growth in the industry, leading to an increased academic interest in the subject. The following sections provide a chronological overview of relevant literature, classified as surveys and broader oriented papers and literature focusing on yard operations respectively.

Two central references stand out: The 2004 survey of OR and container terminals by Steenken et al. [53] and the recent update from 2007 by Stahlbock and Voß [52], together providing a profound practical insight in problems and operations in container terminals as well as a comprehensive overview of literature concerning the subject.

### **Surveys and broader oriented papers**

Surveys of problems occurring in port container terminals and methodologies applied to solve them has been provided by a number of authors. In 1999 Chen [7] described the different terminal operations. In 2001 Meersmans and Dekker [45] focused on OR methods within terminal design and operation. In 2003 Vis and de Koster [59] classified terminal transshipment problems and provided a review of quantitative models and relevant literature for the decision problems. In 2004 Steenken et al. [53] presented a comprehensive overview of the logistic processes in port container terminals and optimization methods for the problems arising within these. In a survey paper from 2007 Stahlbock and Voß [52] update the 2004 review by yet a comprehensive study of OR methods for container terminals problems.

A number of articles have a broad approach to container terminal problems, not focusing on either quayside, yard, or landside operations. In their paper from 1988 dealing with capacity planning, van Hee and Wijbrands [58] introduce models for each main process in the terminal and combine these in a heuristic approach, constituting a decision support system to evaluate the performance of the different port elements. In 1999 Kozan and Preston [38] introduced genetic algorithm techniques to optimize container transfers across marine terminals and the same aggregated problem was addressed in 2000 by Kozan [35], formulating a network flow model to investigate the multimodal terminal and to optimize the container transfer problem. In a 2004 paper Dell’Olmo and Lulli [15] deal with the tactical and operational planning problems occurring in container terminals by viewing the system as a network of platforms for which they formulate a mathematical model and apply dynamic programming based heuristic techniques. In a recent paper from 2007 Kozan and Casey [36] propose simulation and heuristic algorithms for the problem of minimizing ship delays by considering the containers, destined for a given ship, from their arrival in a the port until being loaded onto the ship.

### **Literature concerning container positioning**

Some of the first papers dealing with yard operations in port container terminals are from 1993. Cao and Uebe [5] treat the problem of repositioning containers in a stor-

age block. They formulate the problem of moving containers from certain rows (that must be emptied for arriving containers) to others as a capacitated multicommodity p-median transportation problem, for which they present a Lagrangian relaxation method and a heuristic branch and bound algorithm. Taleb-Ibrahimi et al. [55] apply a strategic/tactical approach to finding the minimal storage space needed to implement a proposed handling and storage strategy given a certain traffic. The paper supports choice of technology and terminal locations at the strategic level, and minimization and prediction of the amount of handling work in the yard at the operational level. As in the paper by Cao and Uebe, the authors work at an aggregated level, not considering details concerning e.g. each container or crane.

In 1994 Kim and Kim [30] presented a quadratic MIP formulation of the problem of assigning storage locations to groups of outbound containers, destined for the same ship. The solution of a small example, using a standard software, serves as an illustration of the algorithm.

Kim and Bae [28] deal with the problem of converting a current storage situation into a required one by moving as few containers as possible, minimizing the total transportation distance. In the paper from 1998, the authors decompose the problem into three subproblems: bay matching, move planning, and task sequencing, for each of which a mathematical model is proposed (but not stated).

The first to use the term “container positioning problem” and to state a mathematical formulation of it were Paolucci et al. [49] in a conference proceedings from 1998 dealing mainly with information and support systems for terminal management. After describing a terminal layout, the decisions to be made, and the requirements for a management system, the authors touch on the problem where inbound containers are to be unloaded and placed at a specific position in the yard such that the total cost of retrieving containers for further transportation is minimized. The problem is formulated as a MIP model, a max cut solution procedure is discussed, but no computational studies are conducted.

A segregation strategy for allocating storage space to inbound containers was revisited and investigated in 1999 by Kim and Kim [31], proposing mathematical models for constant, cyclic, and dynamic arrival rates respectively and solving some test cases to optimality. In the segregation strategy, containers are separated according to their arrival time, such that newly arrived containers are not allowed to be stacked on top of containers already located in the storage area. (Six years earlier, in 1993, this strategy was compared with a storage strategy where the objective was to minimize the variation in container stack height, by de Castilho and Daganzo [14].)

In their paper from 2000 Kim et al. [34] present a dynamic model to minimize the expected number of repositionings when locating outbound containers in the storage yard. To illustrate the algorithm the authors provide a numerical example which is solved to optimality. Realizing that the computation time of the dynamic programming algorithm makes it impractical to apply to larger problem instances, a decision tree approach using the optimal solution set generated by the dynamic programming algorithm is introduced.

In 2001 Preston and Kozan [51] formulated a mathematical model for the container

location problem in order to identify the optimal storage strategy with the objective of minimizing the turn around time for ships in the port. Being NP-hard, the problem is not solved to optimality but a genetic algorithm is applied to solve the proposed model.

Bish et al. [2] extend the container positioning problem to also including vehicle scheduling. They propose an assignment formulation of the problem of assigning containers to storage positions in the yard and dispatching vehicles to containers with the objective of minimizing the time required to unload all containers from a ship berthed in the terminal. Based on the assignment formulation, a heuristic algorithm is presented and analyzed through the solution of a test case. Two years later, in 2003, Bish [1] further extended the problem by adding a third level - the quay crane scheduling. This multiple-crane scheduling and container positioning problem is formulated as a trans-shipment problem, serving as a basis for a heuristic algorithm which is analyzed and tested on a numerical example.

A MIP model for the problem of allocating space to outbound containers is presented by Kim and Park [33]. They develop two heuristic algorithms and compare them through the solution of two test cases.

Also in 2003 a more aggregated approach was suggested by Zhang et al. [63], dealing with the storage space allocation problem by decomposing it into two levels, each formulated as an integer program. At the first level the number of containers allocated to each storage block in the yard is determined, and at the second level the number of containers from/to each ship that is assigned to each storage block is determined. A practical example which illustrates the efficiency of the proposed method is solved.

Chen et al. [6] view the storage space optimization problem as a time and space dimensioned packing problem in their paper from 2004. They propose a number of metaheuristics, such as tabu search, simulated annealing, squeaky wheel optimization (a newer heuristic composed of a construction algorithm, generating elements of a solution, and identification of negatively contributing elements which are handled first by the construction algorithm in the next iteration), and genetic algorithms, to solve it. The objective is to minimize the space required while fulfilling the demand for container storage in the yard. The authors generate data to solve some test cases in order to validate and compare the solution approaches.

In 2006, Kim and Lee [32] investigated the problem of positioning export containers in the yard with the objective of maximizing equipment efficiency by applying the constraint satisfaction technique. The algorithm performance is evaluated by solution of a number of numerical problem instances with data from the Pusan Eastern Container Terminal in Korea.

Kang et al. address the problems of storing containers of uncertain weight and remarshalling export containers destined for the same ship in two papers from 2006. Simulated annealing algorithms are suggested in both papers. The first problem of finding an effective stacking strategy for containers is approached by development of a heuristic which outperforms the traditional approach in a real port terminal by a significant reduction of rehandling [27] and the method presented for the second problem of finding an efficient remarshalling plan is shown promising results by minimizing the

time required for remarshalling and avoiding rehandles during loading and remarshalling [26].

Kozan and Preston [37] propose two models for transferring and locating containers in the terminal yard, combine them in an iterative search algorithm, and apply several metaheuristics to solve large-scale problem instances. The authors find that the proposed genetic algorithm achieves the best results.

Lee et al. [41] introduce a MIP model for the yard storage allocation problem, consisting in transporting containers to and from vessels and allocating storage space for the unloaded containers in the terminal yard, so that congestion and reshuffling is minimized. The model determining the minimum number of yard cranes is not able to solve all numerical examples considered whereas two heuristics developed for the problem outperform previous results.

Kim and Hong [29] consider the problem of determining locations for relocated blocks (containers) by proposing a branch and bound algorithm and a heuristic that estimates the expected number of future relocations for a given stack. A number of test problems serves the purpose of comparing the two approaches, proving the heuristic procedure to outperform the exact method in terms of computation time.

The 2007 paper by Cordeau et al. [9] concerns the tactical service allocation problem, consisting in minimizing the intra-yard container rehandling. The problem is formulated as a generalized quadratic assignment problem with side constraints and two MIP models are stated. As only small problem instances can be solved to optimality, a memetic heuristic algorithm is proposed, providing good results for real-life instances.

Lee and Hsu [42] propose an integer programming formulation of the container pre-marshalling problem, consisting in repositioning export containers so that they can be loaded directly onto the ship. The model, a multi-commodity time-space network flow model with side constraints, is not able to handle larger problem instances for which reason a heuristic procedure is developed, able to solve problems close to real-life sizes.

## Summary

Based on the above literature review, it is believed that this thesis makes new contributions in relation to the modeling and solution of the CPP. Among the papers in which mathematical programming formulations for different versions of the CPP are presented there is a general tendency towards applying a more aggregated approach than suggested in this thesis. Some authors consider groups of containers or allocation of sets of positions and some determine a single position for each container rather than representing a sequence of moves or reshuffles. Furthermore, the time dimension of the lifo principle is not seen modeled which makes the presented models somewhat less complex than if it was included.

A number of papers introduce heuristic algorithms that are able to solve large-scale or real-life problem instances for problems that resemble the CPP as it is defined in this thesis but, again, most approaches are on a more aggregated level.

This thesis introduces highly detailed mathematical optimization models for the CPP and provides a thorough investigation of problem complexity and computational effi-

ciency when solving the models by standard optimization software as well as when relaxing complicating constraints. Furthermore, a robust and flexible heuristic algorithm, capturing all details of the CPP and solving real-life problem instances within few seconds, is presented. This leads to the conclusion that the present thesis provides new and relevant contributions to the research concerning the CPP.

## 1.4 Outline of the thesis

The main part and primary contributions of this thesis are represented by three chapters, 2, 3, and 5, concerning the mathematical models, the heuristic algorithm, and the computational results respectively. The complete thesis is organized as follows.

The present chapter leaves the reader introduced to the thesis in general, the CPP to be addressed, port container terminals and the container industry, and relevant literature concerning the subject. Furthermore, practitioners may have gained insight into scientific aspects of the study and academic readers may have obtained a basic knowledge of the industrial motivation of the thesis.

Chapter 2 is concerned with construction of mathematical programming formulations. Three mathematical optimization models, a first conceptual formulation approach to the entire port container terminal and two models for the CPP are proposed and thorough problem analyses and model comparisons are carried out.

The first model is a MIP model for the entire port container terminal. It considers each container from entering the terminal until departing from it which makes the modeling approach very broad. In this first conceptual formulation is inspired by the well-known vehicle routing problem (VRP). The lifo restrictions account for a very large number of variables and constraints.

The second model is a MIP model as well as the initial one. It captures the CPP except from a number of capacity constraints and, therefore, represents the scope of the thesis as opposed to the first model. The second formulation consists of considerably fewer variables and constraints, especially the lifo restrictions make up a smaller part of the entire model compared to the first approach.

The third model is a BIP model and concerns the CPP but in addition to the second formulation, capacity constraints are included so that the maximum number of simultaneous moves and the stacking height are preserved. Continuous time variables are replaced by time-discretized binary variables, enabling the main advantage of the third model - elimination of “big  $M$ ” terms.

Chapter 3 is concerned with the development of a heuristic algorithm for the CPP. An event-based greedy construction algorithm is proposed and two improvement routines, one embedded in the basic heuristic and one repairing the initial solution, are presented. Equivalently to the models, the object of the approach is to minimize the total transportation time and the total number of moves.

The basic heuristic algorithm constructs a solution by handling a sequence of events, corresponding to required moving operations for containers in the block. Positions for containers are chosen by a number of decision criteria, reflecting the potential advantage

or disadvantage of placing a given container on top of another one in the stack. Two feasibility criteria ensure that no time conflicts will occur when selecting a given position.

Two central issues are addressed in the chapter: 1) Possible rejections of containers for which no feasible positions can be found during one of the events and 2) Delays of departure times for one or several containers. The first issue is dealt with by altering one of the feasibility criteria so that positioning is forced in cases where potential time conflicts with stack containers otherwise prevent it. The second issue constitutes the basis for developing an improvement algorithm that attempts to shift delayed moves to the departure place by exploiting crane idle time previous to the delayed move. Several other perspectives for improving the heuristic solution approach are presented.

The chapter may be more accessible for practitioners than the preceding one concerning the mathematical models. It demonstrates the industrial relevance of the thesis and illustrates an intuitive solution approach to the CPP.

Chapter 4 provides an overview of the computational platforms for implementation and solution of the models as well as the heuristic algorithm and introduces the test problems and real-life data sets used in the solution process. Standard algebraic modeling systems are used for implementation of the mathematical programming models and they are solved by standard MIP solvers. The heuristic is implemented in a standard programming language. A large number of test problems are generated, representing the three types of arrival and departure time patterns for containers that occur in CPP instances. A total of 60 small-scale problem instances of up to 5 containers and 60 medium-scale of up to 10 containers are generated. Furthermore, large real-life data sets from a European port container terminal, used in the performance tests of the heuristic, are described.

Chapter 5 reports the computational results and provides a comprehensive analysis of the outcome and performance of the proposed solution approaches.

The two CPP models are validated and tested by solving the small- and medium-scale test problems. The standard compact model optimization approach with a given run time limit is compared to both solving the models with an extended time limit for the total computation time and solving the models after relaxing a number of complicating constraints. The latter investigation gives some indications of the potential in applying a relaxation approach and adding violated constraints identified by a separation algorithm on the fly.

The small- and medium-scale instances are used for tuning a number of heuristic parameters and serve as a basis for comparison of results from solving the models and the heuristic. Furthermore, a thorough investigation of the heuristic algorithm performance is conducted by solving a wide range of large-scale instances, applying several different strategies.

Model and heuristic results are compared and discussed and some conclusions concerning the suitability of the two solution approaches are presented.

Finally, conclusions and perspectives for future research are given in chapter 6.





## Chapter 2

# New mathematical formulations of container terminal problems

This chapter presents the part of the thesis concerned with constructing mathematical programming formulations. Later chapters are dedicated to solution methods based on the models as well as heuristic approaches.

In order to understand a given problem, it is valuable to construct mathematical models for it. Through the iterative process of modeling, implementing, and testing, the problem structures and properties clarify, facilitating the subsequent process of developing solution methods for the problem. Therefore, the first purpose of building mathematical models for a complex problem is to analyze it. As only few well-structured problems can be solved efficiently by applying standard software to a compact model, it is likely that problem-specific solution methods, such as tailored exact algorithms or heuristics, must be developed for solving large-scale instances.

The starting point for the modeling part of the present research study consists in a broad perspective on port container terminal planning problems. In line with convergence of the study, models for the specific problem, the the container positioning problem (CPP), has been developed.

Altogether, three mathematical optimization models are suggested: An initial conceptual model and two models capturing the CPP. Through different objective functions, the goal of all three models is to minimize the total cost of positioning and reshuffling containers. The initial mixed integer linear programming (MIP) model represents the entire port container terminal - from unloading of vessels on the quayside to loading onto trucks and trains on the landside of the port - and is called the container terminal problem (CTP) model. The second, also a MIP model, captures the CPP as it is described in section 1.1 and is called the CPP model. The third, also representing the CPP, is a pure binary integer linear programming (BIP) model building on a discretization of the time horizon and is called the CPPT model.

The models represent three different approaches to the problem of positioning and reshuffling containers in the terminal. It is not the purpose to compare the models on common terms as they have different strengths and weaknesses. They do not capture the

exact same problems and some features are, for modeling reasons, possible to include in one model but have to be omitted in the others. However, the development in the models reflect the knowledge gain in the process and, thus, the later models represent more efficient formulations than the first one, representing a conceptual modeling approach to the PhD study.

### Common problem properties and model elements

Before describing and analyzing each model in the following sections, common properties are discussed and subsequently some general notation, valid in all three models, is stated. The models differ to some extent but many properties are equivalent. The problems are all modeled in a graph approach where the nodes represent physical places or spots in the terminal in slightly different ways. An issue treated in all three models is the last-in first-out (lifo) principle, described in section 1.1, which is a major challenge to model efficiently. Furthermore, features like arrival and departure times for containers, vessels and vehicles (trucks and trains), and transportation times between nodes in the network are parts of all three models.

It is assumed that arrival and departure times for containers - or the modes of transportation by which they arrive at or depart from the terminal - are known. In practice this assumption is reasonable for vessels since the anchoring times for container ships are typically known, providing decision-makers with rather precise data for large groups of containers. On the other hand, time estimates might be quite uncertain at the landside of the port, as trucks arrive at the terminal within a specified time window but usually not at a specific point in time.

Obvious common properties in the models are the time dimension and the physical layout and restrictions in the port. For any problem instance, the arrival and departure times are given in integers, representing the elapsed time in minutes (or other suitable time units), counting from the beginning of the planning horizon. Between all physical spots in the terminal, a transportation distance can be calculated. In the storage blocks, due to crane safety regulations, moves between positions follow the lines parallel to rows or bays. Therefore, Manhattan ( $L_1$  metric) distances are considered. Correspondingly, transportation times are calculated by summing the lengthwise and crosswise distances and dividing by the crane velocity. All three models include constraints, ensuring the meeting of arrival and departure times, physical restrictions concerning flow and positioning sequences, and observation of the lifo principle.

Common notation for the models includes the set of containers,  $\mathcal{C} = \{1, \dots, C\}$ , the set of positions,  $\mathcal{P} = \{1, \dots, P\}$ , the transportation time between nodes  $(i, j)$  in the network  $T_{ij}$ , and the arrival and departure times,  $A_s$  and  $D_v$  for ship  $s$  and vehicle  $v$  in the initial model, and  $A_c$  and  $D_c$  for container  $c$  in the second and third model. In addition to these basic sets and parameters, some model specific notation is introduced in each of the following sections. Furthermore, throughout all equations in this chapter, the following condensed expressions, exemplified by a set  $\mathcal{Q}$ , are used:  $(q, r) \in \mathcal{Q}^2$ , short for  $q \in \mathcal{Q}, r \in \mathcal{Q}, q \neq r$ ,  $(q < r) \in \mathcal{Q}^2$ , short for  $q \in \mathcal{Q}, r \in \mathcal{Q}, q < r$ , and  $(q > r) \in \mathcal{Q}^2$ , short for  $q \in \mathcal{Q}, r \in \mathcal{Q}, q > r$ .

## 2.1 A MIP model for port container terminals

The first conceptual modeling approach to the problem of handling containers in port terminals, the CTP, takes into account the entire path through the terminal, from the container arrives at the port and is unloaded from the vessel till it is leaving the port by rail or road transportation. This early work served as a basis for developing ideas and improvements, concepts and techniques, resulting in the subsequent models for the CPP. Furthermore, parts of the model, not practicable in its entirety, can be used for future work with the different well-defined problems occurring in the terminal.

The model is inspired by the vehicle routing problem (VRP) where binary decision variables determine the visit pattern and continuous variables control the time dimension. Replacing vehicles, visiting a number of customers in a network, with containers, moving between a number of spots in the terminal, is part of the transformation of the VRP into the CTP.

Figure 2.1 provides a graph representation of the CTP, covering the entire port container terminal.

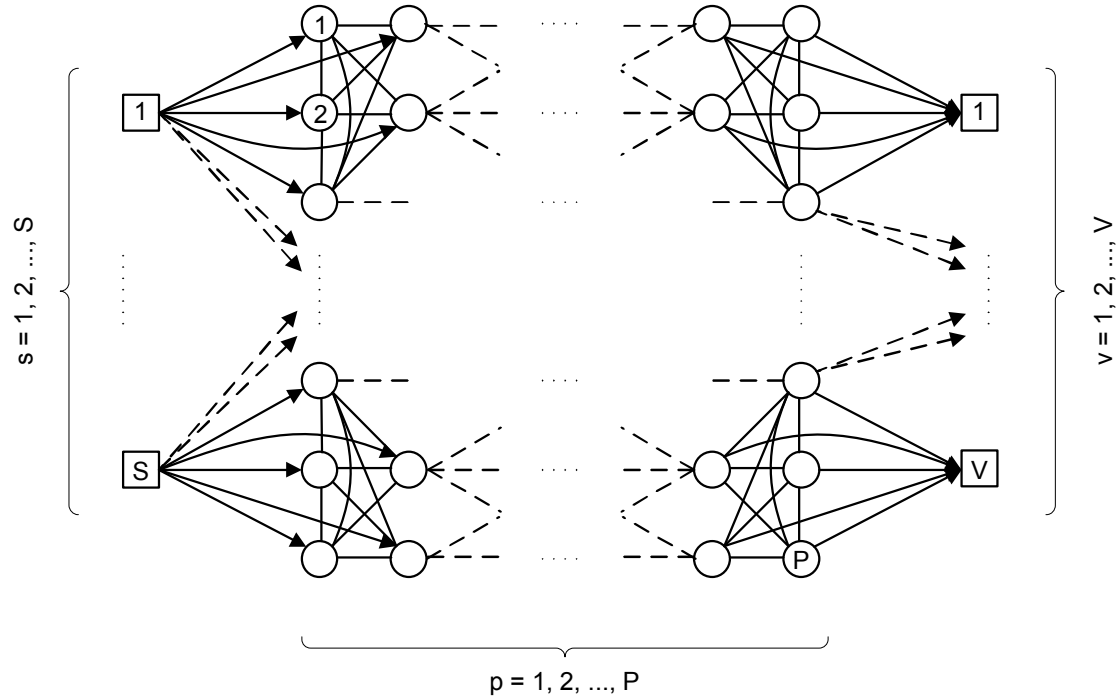


Figure 2.1: Graph representation of the container terminal problem (CTP). Ship nodes are denoted  $s = 1, \dots, S$ , position nodes  $p = 1, \dots, P$ , and vehicle nodes are denoted  $v = 1, \dots, V$ . A set of directed arcs connect all ships with all positions and, correspondingly, a set of directed arcs connect positions with vehicles. All positions within each block are connected with undirected arcs. The illustrated graph is repeated for each container  $c$ , including only the ship and vehicle nodes, compatible with  $c$ .

The CTP model minimizes the total transportation time for containers throughout the terminal, subject to three types of restrictions: Arrival/departure, flow, and lifo constraints. The notation for the model is as follows. The sets,  $\mathcal{C}$  and  $\mathcal{P}$ , and the parameters,  $A_s$ ,  $D_v$ , and  $T_{ij}$ , are described in the chapter introduction. The sets  $\mathcal{S} = \{1, \dots, S\}$  and  $\mathcal{V} = \{1, \dots, V\}$  contain the ships and vehicles respectively and the set  $\mathcal{I} = \mathcal{S} \cup \mathcal{P} \cup \mathcal{V}$  represents all nodes in the network, as ships, positions in the yard, and vehicles, all make nodes in the graph. Figure 2.1 shows a graph representation of the problem.

Furthermore, three tuple sets, representing connections between pairs or groups of containers and transportation means, are introduced.  $\Delta$  contains tuples  $(c, s)$  when container  $c$  arrives on ship  $s$ ,  $\Gamma$  contains tuples  $(c, c', s)$  when container  $c$  is stacked on top of  $c'$  on ship  $s$ , and  $\Upsilon$  contains tuples  $(c, v)$  when container  $c$  and vehicle  $v$  have same destination.

Besides ships' arrival times  $A_s$ , vehicles' departure times  $D_v$ , and node-to-node transportation times  $T_{ij}$ , the parameter  $L = \max\{D_v \mid v \in \mathcal{V}\} - \min\{A_s \mid s \in \mathcal{S}\}$ , equivalent to the length of the planning horizon, is introduced.

Variables in the CTP model are  $x_{cij}$ , controlling the visit pattern,  $t_{ci}$ , associating time stamps with visited nodes in the network, and three variables,  $\sigma_{cc'p}$ ,  $\alpha_{cc'p}$ , and  $\omega_{cc'p}$ , observing the lifo restrictions. The binary  $x_{cij}$  equals 1 if container  $c$  is moved directly from node  $i$  to  $j$ , and 0 otherwise, the continuous  $t_{ci}$  states the time where container  $c$  leaves node  $i$ , and the binary lifo variables indicate different situations for two containers,  $c$  and  $c'$ , at a position  $p$ :  $\sigma_{cc'p}$  equals 1 if container  $c$  and  $c'$  are placed at the same position  $p$ ,  $\alpha_{cc'p}$  equals 1 if container  $c$  is placed at position  $p$  before  $c'$ , and  $\omega_{cc'p}$  equals 1 if there is an overlap between the storage times for container  $c$  and  $c'$  at position  $p$ . Each of the three lifo variables equals 0 if the described situation does not occur. Figure 2.2 illustrates the three variables' role in the lifo principle and table 2.1 provides an overview of the notation for the CTP model.

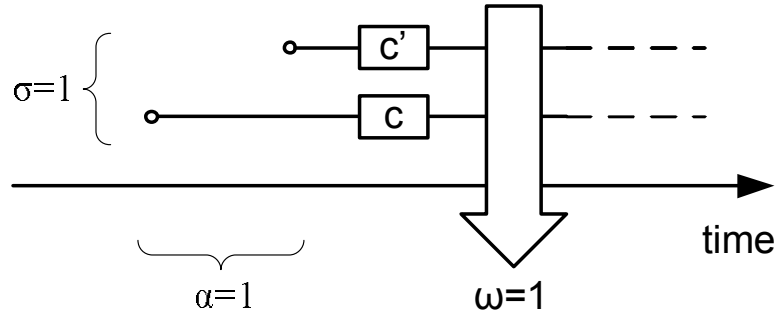


Figure 2.2: Illustration of the lifo principle and the three variables,  $\sigma_{cc'p}$ ,  $\alpha_{cc'p}$ , and  $\omega_{cc'p}$ , in a situation with two containers,  $c$  and  $c'$ , at some position. Both containers being placed at the position implies that  $\sigma_{cc'p} = 1$ , container  $c$  being positioned before container  $c'$  implies that  $\alpha_{cc'p} = 1$ , and the time overlap between the two containers' storage times at the position implies that  $\omega_{cc'p} = 1$ .

| <b>Sets</b>                                                                      |                                                                          |
|----------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| $\mathcal{C} = \{1, \dots, C\}$                                                  | Containers                                                               |
| $\mathcal{S} = \{1, \dots, S\}$                                                  | Container ships                                                          |
| $\mathcal{P} = \{1, \dots, P\}$                                                  | Positions in the yard                                                    |
| $\mathcal{V} = \{1, \dots, V\}$                                                  | Vehicles (trucks and trains)                                             |
| $\mathcal{I} = \{1, \dots, I\}$                                                  | All nodes, $\mathcal{I} = \mathcal{S} \cup \mathcal{P} \cup \mathcal{V}$ |
| $\Delta$ : Tuples $(c, s)$                                                       | Container $c$ arrives on ship $s$                                        |
| $\Gamma$ : Tuples $(c, c', s)$                                                   | Container $c$ is stacked on top of $c'$ on ship $s$                      |
| $\Upsilon$ : Tuples $(c, v)$                                                     | Container $c$ and vehicle $v$ share destination                          |
| <b>Parameters</b>                                                                |                                                                          |
| $A_s, s \in \mathcal{S}$                                                         | Arrival time for ship $s$                                                |
| $D_v, v \in \mathcal{V}$                                                         | Departure time for vehicle (truck/train) $v$                             |
| $T_{ij}, i \in \mathcal{S} \cup \mathcal{P}, j \in \mathcal{P} \cup \mathcal{V}$ | Transportation time between nodes $i$ and $j, i \neq j$                  |
| $L$                                                                              | Length of planning horizon:                                              |
| <b>Decision variables</b>                                                        |                                                                          |
| $x_{cij} \in \mathbb{B}^{C(S+P)(P+V)}$                                           | 1 iff container $c$ is moved directly from node $i$ to $j$               |
| $t_{ci} \in \mathbb{R}_+^{CI}$                                                   | Time of container $c$ leaving node $i$                                   |
| $\sigma_{cc'p} \in \mathbb{B}^{C(C-1)P}$                                         | 1 iff container $c$ and $c'$ are placed at the same position $p$         |
| $\alpha_{cc'p} \in \mathbb{B}^{C(C-1)P}$                                         | 1 iff container $c$ is placed at position $p$ before $c'$                |
| $\omega_{cc'p} \in \mathbb{B}^{C(C-1)P}$                                         | 1 iff container $c$ and $c'$ overlap storage time at position $p$        |

Table 2.1: Notation for the CTP model.

Figure 2.3 provides an overview of the solution structure for the CTP model, exemplified by one container  $c$  being placed at three positions,  $p_1, p_2$ , and  $p_3$ .

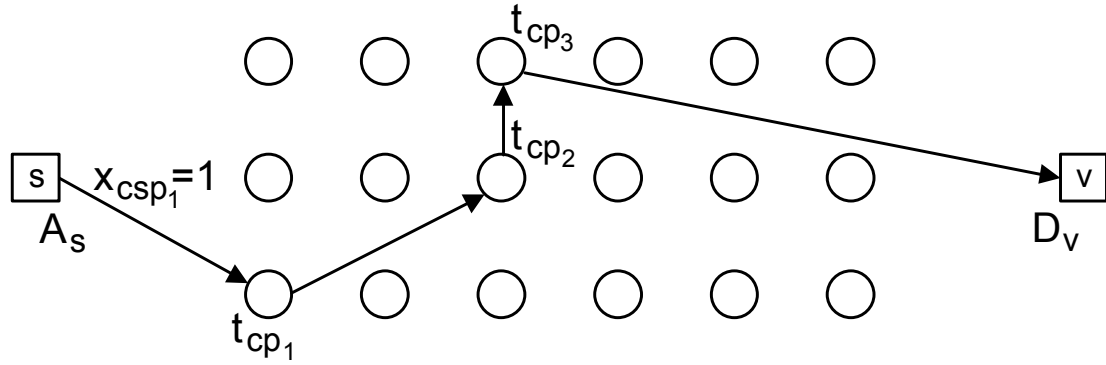


Figure 2.3: Schematic overview of the CTP solution structure. A container  $c$  is placed at three positions,  $p_1, p_2$ , and  $p_3$ , after being unloaded from the ship  $s$  and before being loaded onto the vehicle  $v$ . The edges between nodes represent moves, corresponding to the respective visit variable  $x_{cij}$  being equal to 1. Each node the container visits is associated with a time stamp, corresponding to the continuous variable  $t_{ci}$ .

### 2.1.1 The CTP model

The CTP model minimizes the total transportation time for containers through the terminal subject to 1) Arrival and departure constraints, 2) Flow constraints, and 3) Lifo constraints.

#### Objective

The objective function

$$\sum_{c \in \mathcal{C}, i \in \mathcal{S} \cup \mathcal{P}, j \in \mathcal{P} \cup \mathcal{V}} T_{ij} x_{cij} \quad (2.1.1)$$

sums the transportation times for containers. The total transportation time is minimized in order to reduce the cost of container positioning and reshuffling. Alternatively, to minimize the number of reshuffles directly, the transportation time parameter  $T_{ij}$  can be omitted, leaving the objective function  $\sum_{c \in \mathcal{C}, i \in \mathcal{S} \cup \mathcal{P}, j \in \mathcal{P} \cup \mathcal{V}} x_{cij}$ . The two objectives are correlated but by including the first function, moves over long distances are minimized.

#### Arrival and departure constraints

The first class of constraints (2.1.2) - (2.1.6) concerns unloading of ships and loading of vehicles. Equalities

$$\sum_{p \in \mathcal{P}} x_{csp} = 1, \forall (c, s) \in \Delta \quad (2.1.2)$$

ensure that each container  $c$  is unloaded from the ship  $s$  it has arrived with. Inequalities

$$t_{cs} \geq A_s, \forall (c, s) \in \Delta \quad (2.1.3)$$

restrict the earliest unloading time for all containers  $c$  arriving with ship  $s$  to the arrival time for the vessel. The last unloading constraints

$$t_{cs} + \sum_{p \in \mathcal{P}} T_{sp} x_{csp} \leq t_{c's}, \forall (c, c', s) \in \Gamma \quad (2.1.4)$$

ensure the right unloading sequence for each ship by restricting the order of the time variables,  $t_{cs}$  and  $t_{c's}$ , for every pair of containers,  $c$  and  $c'$ , where  $c$  is stacked on top of  $c'$  on ship  $s$ . Equalities

$$\sum_{p \in \mathcal{P}, v \in \mathcal{V}} x_{cpv} = 1, \forall (c, v) \in \Upsilon \quad (2.1.5)$$

match each container  $c$  with a vehicle  $v$ , destined for the zone  $z$  where  $c$  is to be delivered to, and in inequalities

$$t_{cv} \leq D_v \sum_{p \in \mathcal{P}} x_{cpv}, \forall (c, v) \in \Upsilon \quad (2.1.6)$$

the latest loading time for container  $c$ , leaving the port by vehicle  $v$ , is set to be the departure time for  $v$ . If  $c$  is not matched with  $v$ , i.e. if  $\sum_{p \in \mathcal{P}} x_{cpv} = 0$ , the time variable  $t_{cv}$  is unchanged from the initial value of 0.

### Flow constraints

Three constraints, (2.1.7), (2.1.8), and (2.1.9), concern flow restrictions. Equalities

$$\sum_{i \in \mathcal{S} \cup \mathcal{P}} x_{cip} = \sum_{j \in \mathcal{P} \cup \mathcal{V}} x_{cpj}, \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (2.1.7)$$

preserve the node balance for positions in the yard by ensuring that if container  $c$  is placed at position  $p$  it is removed from the position again. Correspondingly, inequalities

$$t_{ci} + T_{ij}x_{cij} \leq t_{cj} + L(1 - x_{cij}), \quad \forall c \in \mathcal{C}, i \in \mathcal{S} \cup \mathcal{P}, j \in \mathcal{P} \cup \mathcal{V} \quad (2.1.8)$$

preserve the time balance for nodes in the network by restricting the earliest time  $t_{cj}$  where container  $c$  can leave node  $j$ . The connection between the binary and the continuous variables is formulated in inequalities

$$t_{cp} + \sum_{c' \in \mathcal{C}} (\sigma_{cc'p} + \alpha_{cc'p} + \omega_{cc'p}) \leq (L + 3) \sum_{j \in \mathcal{P} \cup \mathcal{V}} x_{cpj}, \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (2.1.9)$$

ensuring that  $x_{cpj}$  dominate the other decision variables,  $t_{cp}$ ,  $\sigma_{cc'p}$ ,  $\alpha_{cc'p}$ , and  $\omega_{cc'p}$ , by setting them to 0 if  $x_{cpj} = 0$ .

### Lifo constraints

Constraints (2.1.10) - (2.1.17) concern the lifo principle. The first six constraints determine the binary decision variables,  $\sigma_{cc'p}$ ,  $\alpha_{cc'p}$ , and  $\omega_{cc'p}$ , and in the last two constraints, these are used to define the lifo restrictions. In inequalities

$$\sigma_{cc'p} \geq \sum_{i \in \mathcal{S} \cup \mathcal{P}} (x_{cip} + x_{c'ip}) - 1, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.10)$$

the binary variable  $\sigma_{cc'p}$  is set to 1 if both container  $c$  and  $c'$  are placed at position  $p$ . The property that the value of  $\sigma_{cc'p}$  is the same for the pairs  $(c, c')$  and  $(c', c)$  is formulated in

$$\sigma_{cc'p} = \sigma_{c'cp}, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.11)$$

where  $\sigma_{cc'p}$  is mirrored, fixing  $\sigma_{c'cp}$  to the same value. The connection between  $\sigma_{cc'p}$  and the binary variable  $\alpha_{cc'p}$ , indicating the first positioned of two containers,  $c$  and  $c'$ , at a position  $p$ , is formulated in equalities

$$\sigma_{cc'p} = \alpha_{cc'p} + \alpha_{c'cp}, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.12)$$

setting either  $\alpha_{cc'p}$  or  $\alpha_{c'cp} = 1$  if  $\sigma_{cc'p} = 1$  and  $\alpha_{cc'p} = \alpha_{c'cp} = 0$  if  $\sigma_{cc'p} = 0$  respectively, and by inequalities

$$\sum_{i \in \mathcal{S} \cup \mathcal{P}} (t_{ci} + T_{ip}x_{cip}) \leq \sum_{i \in \mathcal{S} \cup \mathcal{P}} (t_{c'i} + T_{ip}x_{c'ip}) + L(1 - \alpha_{cc'p}), \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.13)$$



$\alpha_{cc'p}$  is set to 0 if container  $c$  is placed at position  $p$  after container  $c'$ . The binary variable  $\omega_{cc'p}$ , representing a situation with time overlap of two containers,  $c$  and  $c'$ , at a position  $p$ , is determined in

$$\sum_{i \in \text{SUP}} (t_{c'i} + T_{ip}x_{c'ip}) \geq t_{cp} - L(1 - \alpha_{cc'p} + \omega_{cc'p}), \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.14)$$

where  $\omega_{cc'p}$  is forced to 1 if container  $c$  is placed at position  $p$  before container  $c'$  (i.e.  $\alpha_{cc'p} = 1$ ) and  $c$  is removed from  $p$  after  $c'$  is placed at the position. Like in equations (2.1.11), below inequalities formulate the symmetry of  $\omega_{cc'p}$  and  $\omega_{c'cp}$

$$\omega_{cc'p} = \omega_{c'cp}, \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.15)$$

by mirroring the variables. The following two constraints, building on  $\alpha_{cc'p}$  and  $\omega_{cc'p}$ , preserve the life restrictions. One of them is enforced for a pair of containers,  $c$  and  $c'$ , when  $\alpha_{cc'p} = 1$ . Which one becomes active depends on whether there is a time overlap between the storage times for the containers at the position (i.e. when  $\omega_{cc'p} = 1$ ) or the two containers are placed at the position in separate time intervals (i.e. when  $\omega_{cc'p} = 0$ ). In the constraints

$$t_{cp} \geq t_{c'p} + \sum_{j \in \mathcal{P} \cup \mathcal{V}} T_{pj}x_{c'pj} - L(2 - (\alpha_{cc'p} + \omega_{cc'p})), \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.16)$$

concerning the first situation, the last term in the inequality's right hand side disappears when  $\alpha_{cc'p} = \omega_{cc'p} = 1$ , thereby, forcing container  $c$  to be removed from position  $p$  after container  $c'$  if  $c$  was placed at  $p$  first and there is an overlap between their storage times. The second situation is accounted for in the inequalities

$$\sum_{i \in \text{SUP}} (t_{c'i} + T_{ip}x_{c'ip}) \geq t_{cp} + \sum_{j \in \mathcal{P} \cup \mathcal{V}} T_{pj}x_{cpj} - L(1 - \alpha_{cc'p} + \omega_{cc'p}), \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.17)$$

which become binding when  $\alpha_{cc'p} = 1$  and  $\omega_{cc'p} = 0$ , eliminating the last term and, thereby, forcing container  $c$  to be removed from position  $p$  before container  $c'$  is placed at  $p$  if  $c$  was the first to be positioned at  $p$  and there is no overlap between their storage times.

### The complete CTP model

As described in the section introduction, the goal of the CTP model is to minimize the total cost of positioning and reshuffling containers in the terminal by shortest possible transportation times, subject to observation of the problem restrictions. Consequently, with the variables domains (2.1.18), the CTP minimization model can be stated as:

$$\text{minimize } \sum_{c \in \mathcal{C}, i \in \mathcal{S} \cup \mathcal{P}, j \in \mathcal{P} \cup \mathcal{V}} T_{ij} x_{cij} \quad (2.1.1)$$

subject to

$$\sum_{p \in \mathcal{P}} x_{csp} = 1, \quad \forall (c, s) \in \Delta \quad (2.1.2)$$

$$t_{cs} \geq A_s, \quad \forall (c, s) \in \Delta \quad (2.1.3)$$

$$t_{cs} + \sum_{p \in \mathcal{P}} T_{sp} x_{csp} \leq t_{c's}, \quad \forall (c, c', s) \in \Gamma \quad (2.1.4)$$

$$\sum_{p \in \mathcal{P}, v \in \mathcal{V}} x_{cpv} = 1, \quad \forall (c, v) \in \Upsilon \quad (2.1.5)$$

$$t_{cv} \leq D_v \sum_{p \in \mathcal{P}} x_{cpv}, \quad \forall (c, v) \in \Upsilon \quad (2.1.6)$$

$$\sum_{i \in \mathcal{S} \cup \mathcal{P}} x_{cip} = \sum_{j \in \mathcal{P} \cup \mathcal{V}} x_{cpj}, \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (2.1.7)$$

$$t_{ci} + T_{ij} x_{cij} \leq t_{cj} + L(1 - x_{cij}), \quad \forall c \in \mathcal{C}, i \in \mathcal{S} \cup \mathcal{P}, j \in \mathcal{P} \cup \mathcal{V} \quad (2.1.8)$$

$$t_{cp} + \sum_{c' \in \mathcal{C}} (\sigma_{cc'p} + \alpha_{cc'p} + \omega_{cc'p}) \leq (L + 3) \sum_{j \in \mathcal{P} \cup \mathcal{V}} x_{cpj}, \quad \forall c \in \mathcal{C}, p \in \mathcal{P} \quad (2.1.9)$$

$$\sigma_{cc'p} \geq \sum_{i \in \mathcal{S} \cup \mathcal{P}} (x_{cip} + x_{c'ip}) - 1, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.10)$$

$$\sigma_{cc'p} = \sigma_{c'cp}, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.11)$$

$$\sigma_{cc'p} = \alpha_{cc'p} + \alpha_{c'cp}, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.12)$$

$$\sum_{i \in \mathcal{S} \cup \mathcal{P}} (t_{ci} + T_{ip} x_{cip}) \leq \sum_{i \in \mathcal{S} \cup \mathcal{P}} (t_{c'i} + T_{ip} x_{c'ip}) + L(1 - \alpha_{cc'p}), \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.13)$$

$$\sum_{i \in \mathcal{S} \cup \mathcal{P}} (t_{c'i} + T_{ip} x_{c'ip}) \geq t_{cp} - L(1 - \alpha_{cc'p} + \omega_{cc'p}), \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.14)$$

$$\omega_{cc'p} = \omega_{c'cp}, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.15)$$

$$t_{cp} \geq t_{c'p} + \sum_{j \in \mathcal{P} \cup \mathcal{V}} T_{pj} x_{c'pj} - L(2 - (\alpha_{cc'p} + \omega_{cc'p})), \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.16)$$

$$\sum_{i \in \mathcal{S} \cup \mathcal{P}} (t_{c'i} + T_{ip} x_{c'ip}) \geq t_{cp} + \sum_{j \in \mathcal{P} \cup \mathcal{V}} T_{pj} x_{cpj} - L(1 - \alpha_{cc'p} + \omega_{cc'p}), \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.1.17)$$

$$x_{cij} \in \mathbb{B}^{C(S+P)(P+V)}; \quad t_{ci} \in \mathbb{R}^{CI}; \quad \sigma_{cc'p}, \alpha_{cc'p}, \omega_{cc'p} \in \mathbb{B}^{C(C-1)P} \quad (2.1.18)$$

### 2.1.2 Model analysis

The CTP model is polynomial with a order of  $\mathcal{O}(C^2P)$  variables and  $\mathcal{O}(C^2P)$  constraints.

The size of the sets,  $\mathcal{C}$ ,  $\mathcal{I} = \mathcal{S} \cup \mathcal{P} \cup \mathcal{V}$ , and the tuple sets,  $\Delta$ ,  $\Gamma$ , and  $\Upsilon$ , determine the number of the variables and constraints. It can be assumed that  $S \ll P$ ,  $P \leq V \leq C$ , and  $2 \leq C/P \leq 3$  for a planning period of about one week. The upper bounds for the number of elements in the tuple sets are as follows. Clearly,  $|\Delta| = C$  since every container  $c$  arrives by exactly one ship  $s$ . The size of  $\Delta$  does not vary for different problem instances. The worst-case size of  $\Gamma$  corresponds to the hypothetical situation where all containers  $c$  are stacked on top of each other on one ship  $s$ , implying  $|\Gamma| \leq (C-1) + (C-2) + \dots + 1$ . For  $\Upsilon$ , the upper bound  $|\Upsilon| \leq C \cdot V$ , is reached in the hypothetical situation where all containers and all vehicles are destined for the same end point, i.e. where each container  $c$  can be assigned to an arbitrary vehicle  $v$ . In reality, the sets,  $\Gamma$  and  $\Upsilon$ , are much smaller.

With  $x_{cij} \in \{0, 1\}$ ,  $\forall c \in \mathcal{C}$ ,  $i \in \mathcal{S} \cup \mathcal{P}$ ,  $j \in \mathcal{P} \cup \mathcal{V}$  corresponding to  $C(S+P)(P+V)$  visit variables,  $t_{ci} \in \mathbb{R}_+$ ,  $\forall c \in \mathcal{C}$ ,  $i \in \mathcal{I}$  corresponding to  $CI = C(S+P+V)$  time variables, and  $\sigma_{cc'p}, \alpha_{cc'p}, \omega_{cc'p} \in \{0, 1\}$ ,  $\forall c \in \mathcal{C}$ ,  $c' \in \mathcal{C}$ ,  $c \neq c'$ ,  $p \in \mathcal{P}$  corresponding to  $3C(C-1)P \simeq 3C^2P$  lifo variables, the upper bound for the total number of variables is  $(CPV + CP^2 + CSV + CSP) + (CV + CP + CS) + 3C^2P$ , with  $(CPV + CP^2 + CSV + CSP) + 3C^2P$  of them being binary. The dominant term is  $C^2P$ , occasioned by the binary lifo variables,  $\sigma_{cc'p}$ ,  $\alpha_{cc'p}$ , and  $\omega_{cc'p}$ .

The arrival and departure restrictions constitute  $2C + \sum_{k=1}^{C-1} (C-k) + 2CV$  constraints, flow restrictions number  $2CP + C(S+P)(P+V)$  constraints, and the lifo restrictions add another  $8C(C-1)P \simeq 8C^2P$  equations and inequalities. Leaving out negligible terms, such as  $-1$ , this gives rise to an upper bound of  $8C^2P + CPV + CP^2 + CSV + CSP + 2CV + 2CP + 2C + \sum_{k=1}^{C-1} (C-k)$  for the total number of constraints. Again, the lifo principle represents the dominant term  $C^2P$ .

A small but realistic problem instance with a weekly throughput of 300 containers arriving on 5 ships, 10 storage blocks with 20 positions each, and 250 vehicles makes a problem instance with approximately 82 million variables, 99.8 % of them binary and 66 % of them concerning the lifo restrictions, and about 172 million constraints, of which 144 million, corresponding to 83.7 %, concern the lifo restrictions. Clearly, the lifo principle gives rise to a large number of variables and constraints. Developing more efficient ways to observe the lifo restrictions and eliminating some of the binary variables and constraints will, therefore, be likely to aid the solution process.

Serving as a preliminary modeling approach to the entire port container terminal the CTP model is implemented in GAMS and validated by solving a small test case which is documented in appendix C. Henceforward, the focus is on handling containers in the storage blocks - the positioning problem - and the CTP model is left at the validation phase, remaining the conceptual basis for constructing the two models, suggested in the following sections, the CPP and CPPT models.

## 2.2 A MIP model for the CPP

The second modeling approach captures the CPP as it is described in section 1.1. Compared to the CTP model the perspective is narrowed down to considering only one storage block at a time. Consequently, the CPP model observes and determines moves from containers arrive at the block till they depart from the block, i.e. the quayside transportation is omitted. This approach implies two additional assumptions: 1) The port has a predetermined strategy for positioning containers in certain blocks. This assumption is reasonable since containers of different types, e.g. reefer containers and oversized boxes, are placed in different blocks. Moreover, 20' and 40' containers are not placed at the same positions within a block which further reduces the number of potential positions for each container. 2) Containers are not moved between blocks, i.e. each container is only repositioned within its block. This assumption is also fairly reasonable considering the cost of transportation across blocks which involves several different types of equipment whereas moves within a single block are carried out by the block crane only.

The CPP model determines optimal positions of containers in one block by minimizing the total number of reshuffles and the total transportation time between positions. Equivalent to the CTP model three types of restrictions are observed: Arrival/departure, flow, and lifo constraints. The notation for the model is as follows. The sets,  $\mathcal{C}$  and  $\mathcal{P}$ , and the parameters,  $A_c$ ,  $D_c$ , and  $T_{ij}$ , are described in the chapter introduction. The edge variable  $x_{cij}$  in the CTP model is changed into a node variable, denoting the position for a container after a certain number of reshuffles. For this purpose, the set  $\mathcal{N} = \{1, \dots, N\}$  is introduced, containing the numbers of reshuffles a container can undergo, e.g.  $n = 1, 2, 3$ , representing three different positions for a container. In addition to the set of positions  $\mathcal{P}$  an initial arrival place and a final departure place is used, as described in section 1.1. These two are considered positions at each end of the block and are denoted  $i = 0$  and  $i = I$ . The set  $\mathcal{I}$  contains all nodes in the network, the arrival place  $i = 0$ , the positions  $i = 1, \dots, P$  where  $P = I - 1$ , and the departure place  $i = I$ , i.e.  $\mathcal{I} = \mathcal{P} \cup \{0, I\} = \{0, 1, \dots, P, I\}$ . Figure 2.4 illustrates the CPP model graph.

Furthermore, the sets  $\mathcal{P}_0 = \mathcal{P} \cup \{0\}$  and  $\mathcal{P}_I = \mathcal{P} \cup \{I\}$  are introduced. Reshuffle number  $n = 1$ , i.e. the first position, corresponds to  $i = 0$  and  $n \geq 2$  corresponds to actual positions in the block,  $i = 1, \dots, P$ . However, for  $i = I$ ,  $n \geq 3$ , since there is at least one actual position in the block.

Besides the parameters, described in the chapter introduction - containers' arrival and departure times,  $A_c$  and  $D_c$ , and node-to-node transportation times  $T_{ij}$  - the CPP model also contains the parameter  $L = \max\{D_c \mid c \in \mathcal{C}\} - \min\{A_c \mid c \in \mathcal{C}\}$ , equal to the length of the planning horizon.

The binary variable  $x_{cni}$  equals 1 if container  $c$  is placed at node  $i$  after  $n$  reshuffles, and 0 otherwise. The continuous decision variables,  $t_{cni}^p$ ,  $t_{cni}^s$ , and  $t_{cni}^m$ , represent positioning times (points in time), storage, and moving times (time intervals), associated with the sequence of positions for each container. Finally, the binary variables,  $\alpha_{cc'i}$  and  $\delta_{cc'i}$ , serve in the preservation of the lifo principle, restricting the continuous time

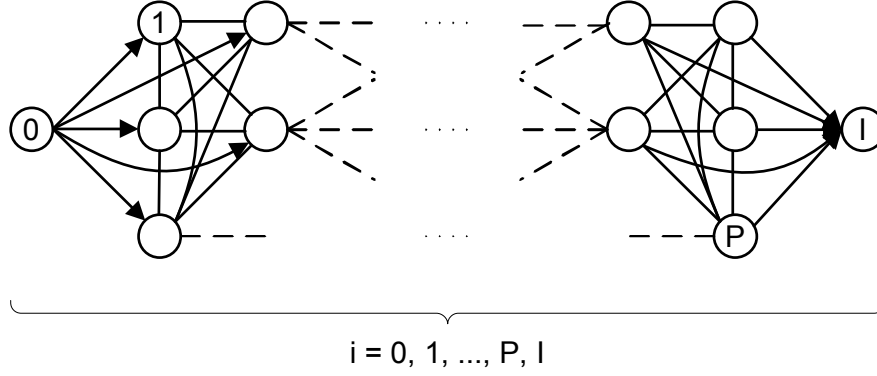


Figure 2.4: The CPP graph for each container  $c$ , consisting of nodes  $i = 0, 1, \dots, P, I$ , with  $i = 0$  and  $i = I$  corresponding to the arrival and departure places and  $i = 1, \dots, P = \mathcal{P}$  representing the positions in the block. Directed arcs connect the arrival place with all positions which are again connected to the departure place with directed arcs. All positions are connected with undirected arcs. For each potential reshuffle  $n$ , the network is repeated.

variables if two containers are placed at the same position, i.e. their respective  $x_{cni}$  variables both equal 1.

The following must be assured: If container  $c$  and  $c'$  are both placed at position  $p$ , and if container  $c$  is positioned before container  $c'$  and is removed from the position after container  $c'$  is placed at  $p$  (i.e. there is a time overlap between the storage times for the two containers at the position), then container  $c$  must be removed after container  $c'$ . This can be formulated as the following implication (with  $\sum_n x_{cnp} = \sum_n x_{c'np} = 1$ , corresponding to both container  $c$  and  $c'$  being placed at position  $p$ ),

$$\left. \begin{array}{l} \sum_n t_{cnp}^p < \sum_n t_{c'np}^p \\ \sum_n t_{cnp}^p + \sum_n t_{cnp}^s > \sum_n t_{c'np}^p \end{array} \right\} \Rightarrow \sum_n t_{cnp}^p + \sum_n t_{cnp}^s > \sum_n t_{c'np}^p + \sum_n t_{c'np}^s$$

The strict inequalities ensure that containers are not placed at or removed from a position simultaneously.

The set  $\mathcal{Z} = \mathcal{C} \times \mathcal{N} \times \mathcal{I}$  is introduced, where  $z = (c, n, i) \in \mathcal{Z}$ . In order to achieve a clear and condensed model the following notation is introduced, exemplified with the tuple index  $q$  and the set  $\mathcal{Q}$ : let  $q = (\pi, \phi) = (q_\pi, q_\phi) \in \mathcal{Q} = \Pi \times \Phi$ , then  $x(\mathcal{Q}) = \sum_{q \in \mathcal{Q}} x_q$  and  $\mathcal{Q}_\pi = \{q = (\pi', \phi) \in \mathcal{Q} \mid \pi' = \pi\}$ , i.e.  $x(\mathcal{Q}_\pi) = \sum_{\phi \in \Phi} x_{(\pi, \phi)}$ ,  $\forall \pi$ . Likewise, the macros  $t^p(\mathcal{Q})$ ,  $t^s(\mathcal{Q})$  and  $t^m(\mathcal{Q})$  are used. Furthermore, in constraints enforced for a subset of elements in a set, the following notation, exemplified with the set  $\mathcal{Q} = \{q_1, q_2, \dots, q_{n-1}, q_n\}$ , is introduced:  $\mathcal{Q}^{k \rightarrow} = \mathcal{Q} \setminus \{q_1, q_2, \dots, q_k\}$  and  $\mathcal{Q}^{\rightarrow k} = \mathcal{Q} \setminus \{q_{n-k+1}, q_{n-k+2}, \dots, q_n\}$ .

Table 2.2 provides an overview of the notation for the CPP model and figure 2.5, showing part of a sequence of positions for one container  $c$ , provides an overview of the solution structure of the CPP presented.

|                                                    |                                                                              |
|----------------------------------------------------|------------------------------------------------------------------------------|
| <b>Sets</b>                                        |                                                                              |
| $\mathcal{C} = \{1, \dots, C\}$                    | Containers                                                                   |
| $\mathcal{N} = \{1, \dots, N\}$                    | Numbers of reshuffles                                                        |
| $\mathcal{P} = \{1, \dots, P\}$                    | Positions                                                                    |
| $\mathcal{P}_0 = \{0, 1, \dots, P\}$               | Positions $\mathcal{P}$ and arrival place $\{0\}$                            |
| $\mathcal{P}_I = \{1, \dots, P, I\}$               | Positions $\mathcal{P}$ and departure place $\{I\}$                          |
| $\mathcal{I} = \{0, 1, \dots, P, I\}$              | All nodes: Positions $\mathcal{P}$ , arrival and departure places $\{0, I\}$ |
| <b>Parameters</b>                                  |                                                                              |
| $A_c, c \in \mathcal{C}$                           | Arrival time for container $c$                                               |
| $D_c, c \in \mathcal{C}$                           | Departure time for container $c$                                             |
| $T_{ij}, i \in \mathcal{P}_0, j \in \mathcal{P}_I$ | Transportation time between nodes $i$ and $j, i \neq j$                      |
| $L$                                                | Length of planning horizon:                                                  |
| <b>Decision variables</b>                          |                                                                              |
| $x_z \in \mathbb{B}^{ \mathcal{Z} }$               | 1 iff container $c$ is placed at position $i$ after $n$ reshuffles           |
| $t_z^p \in \mathbb{R}_+^{ \mathcal{Z} }$           | Positioning time for container $c$ at node $i$ after $n$ reshuffles          |
| $t_z^s \in \mathbb{R}_+^{ \mathcal{Z} }$           | Storage time for container $c$ at node $i$ after $n$ reshuffles              |
| $t_z^m \in \mathbb{R}_+^{ \mathcal{Z} }$           | Moving time for container $c$ after node $i$ and $n$ reshuffles              |
| $\alpha_{cc'p} \in \mathbb{B}^{C(C-1)P}$           | 1 iff container $c$ is placed at position $p$ before $c'$                    |
| $\delta_{cc'p} \in \mathbb{B}^{C(C-1)P}$           | 1 iff container $c$ is removed from position $p$ after $c'$ is placed        |

Table 2.2: Notation for the CPP model.

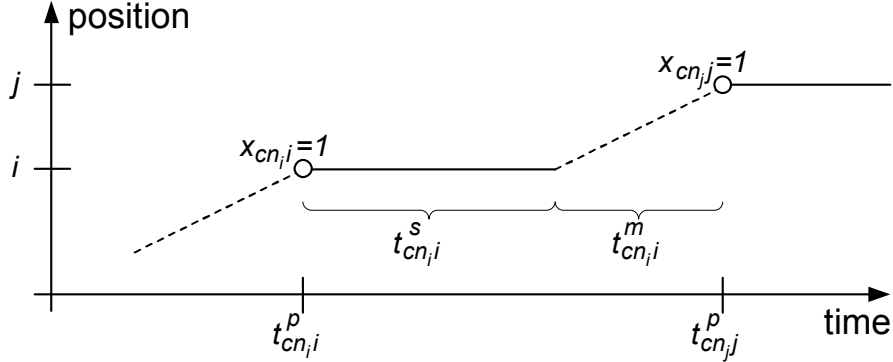


Figure 2.5: Schematic overview of the CPP solution structure: Part of a sequence of positions for one container  $c$  moving from position  $i$  to  $j$ . The binary decision variables,  $x_{cn,i}$  and  $x_{cn,j}$ , both equal 1, indicating that container  $c$  is placed at position  $i$  after  $n_i$  reshuffles and at position  $j$  after  $n_j (= n_i + 1)$  reshuffles. The container's positioning times at the two positions are  $t_{cn,i}^p$  and  $t_{cn,j}^p$  respectively, and the time intervals,  $t_{cn,i}^s$  and  $t_{cn,i}^m$ , correspond to the storage (horizontal line) and moving (sloped line) times between the two positions.

### 2.2.1 The CPP model

The CPP model minimizes the total number of reshuffles and the total transportation time in the storage block subject to 1) Arrival and departure constraints, 2)

Flow constraints, and 3) Lifo constraints.

## Objective

The objective function

$$x(\mathcal{C} \times \mathcal{N} \times \mathcal{P}) + t^m(\mathcal{Z}) \quad (2.2.1)$$

sums the number of positions assigned to all containers plus the total transportation time. The CPP model targets the minimal number of reshuffles in the block and total time spend on moving containers between positions.

## Arrival and departure constraints

The first two equations, stated below, ensure that arrival and departure times for each container are met.

$$t_{c10}^p = A_c, \forall c \in \mathcal{C} \quad (2.2.2)$$

$$t^p(\mathcal{Z}_{ci} \mid n > 2, i = \mathcal{I}) = D_c, \forall c \in \mathcal{C} \quad (2.2.3)$$

In (2.2.2) the first positioning time is set equal to the arrival time for every container  $c$  and, correspondingly, equalities (2.2.3) set the positioning time at the departure place.

## Flow constraints

The majority of the constraints (2.2.4) - (2.2.11) concern the flow restrictions. The inequalities

$$x(\mathcal{Z}_{cn}) \geq x(\mathcal{Z}_{cn+1}), \forall c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1} \quad (2.2.4)$$

ensure the continuity of a sequence of positions by forcing  $x(\mathcal{Z}_{cn}) = 1$  if  $x(\mathcal{Z}_{cn+1}) = 1$ . Besides observing this obvious feasibility property of the solution structure, inequalities (2.2.4) play a role in other constraints such as the following. The moving times between actual pairs of positions are determined by (2.2.5) and (2.2.6) below, both enforced for  $c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I$

$$t_{cni}^m \geq T_{ij}(x_{cni} + x_{cn+1j} - 1) \quad (2.2.5)$$

$$t_{cni}^m \leq T_{ij}(x_{cni} + x_{cn+1j} - 1) + L(2 - x_{cni} - x_{cn+1j}) \quad (2.2.6)$$

setting the lower and the upper bound on  $t_{cni}^m$  respectively. The constraints are only enforced when container  $c$  is moved directly from position  $i$  to  $j$ , i.e. when  $x_{cni} = x_{cn+1j} = 1$  for some  $n$ . When this is not the case, the right-hand side of inequalities (2.2.5) equals 0 or  $-T_{ij}$  and the last part of inequalities (2.2.6) equals  $L$  or  $2L$ , making the constraints non-binding. The same technique is used in inequalities (2.2.7) and (2.2.8), enforced for  $c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I$

$$t_{cni}^p + t_{cni}^s + t_{cni}^m \leq t_{cn+1j}^p + L(2 - x_{cni} - x_{cn+1j}) \quad (2.2.7)$$

$$t_{cni}^p + t_{cni}^s + t_{cni}^m \geq t_{cn+1j}^p - L(2 - x_{cni} - x_{cn+1j}) \quad (2.2.8)$$

ensuring the balance between the positioning, storage and moving times. As for the constraints (2.2.5) and (2.2.6), either both inequalities (2.2.7) and (2.2.8) are binding (i.e.  $x_{cni} = x_{cn+1j} = 1$ ) or both are non-binding. In inequalities

$$t_z^p + t_z^s + t_z^m \leq 3Lx_z, \quad \forall z \in \mathcal{Z} \quad (2.2.9)$$

the connection between the binary  $x_z$  and the continuous,  $t_z^p$ ,  $t_z^s$  and  $t_z^m$ , ensures that the time variables are not assigned values larger than 0 if the corresponding design variable equals 0. The two inequalities

$$x(\mathcal{Z}_{cn}) \leq 1, \quad \forall c \in \mathcal{C}, n \in \mathcal{N} \quad (2.2.10)$$

$$x(\mathcal{Z}_{ci}) \leq 1, \quad \forall c \in \mathcal{C}, i \in \mathcal{I} \quad (2.2.11)$$

imply that container  $c$  is not assigned more than one node  $i$  at a time (i.e. per number  $n$ ) and does not return to a node, previously visited. The latter restriction is included for modeling reasons.

### Lifo constraints

Constraints (2.2.12) - (2.2.16) concern the lifo restrictions, all enforced for every pair of containers  $(c, c') \in \mathcal{C}^2$  and positions  $p \in \mathcal{P}$ . A “big  $M$ ”-modeling technique is used to control the two lifo variables,  $\alpha_{cc'p}$  and  $\delta_{cc'p}$ , in situations where the three lifo conditions hold: 1) Both container  $c$  and  $c'$  are placed at position  $p$  at some point in time, 2) container  $c$  is positioned before container  $c'$ , and 3) container  $c$  is removed after container  $c'$  is placed.  $1/L$ , i.e. a very small value compared to the length of the planning horizon, is used to set the respective variables equal to 1 if relations between the continuous time variables give reason to do so. Consequently, inequalities

$$\alpha_{cc'p} \geq x(\mathcal{Z}_{cp}) + x(\mathcal{Z}_{c'p}) + [t^p(\mathcal{Z}_{c'p}) - t^p(\mathcal{Z}_{cp})]/L - 2 \quad (2.2.12)$$

set  $\alpha_{cc'p} = 1$  if container  $c$  is placed at position  $p$  before container  $c'$ , i.e.  $t^p(\mathcal{Z}_{c'p}) > t^p(\mathcal{Z}_{cp})$ , and inequalities

$$\delta_{cc'p} \geq x(\mathcal{Z}_{cp}) + x(\mathcal{Z}_{c'p}) + [t^p(\mathcal{Z}_{cp}) + t^s(\mathcal{Z}_{cp}) - t^p(\mathcal{Z}_{c'p})]/L - 2 \quad (2.2.13)$$

set  $\delta_{cc'p} = 1$  if container  $c$  is removed from position  $p$  after container  $c'$  is placed at  $p$ , i.e.  $t^p(\mathcal{Z}_{cp}) + t^s(\mathcal{Z}_{cp}) > t^p(\mathcal{Z}_{c'p})$ . To ensure that containers  $c$  and  $c'$  are not placed at position  $p$  simultaneously, the two constraints

$$\alpha_{cc'p} + \alpha_{c'cp} \leq 1 \quad (2.2.14)$$

$$t^p(\mathcal{Z}_{cp}) + 1/L \leq t^p(\mathcal{Z}_{c'p}) + L(1 - \alpha_{cc'p}) \quad (2.2.15)$$

restrict at most one of  $\alpha_{cc'p}$  and  $\alpha_{c'cp}$  to equal one and, given  $\alpha_{cc'p} = 1$ , that container  $c$  is actually placed at position  $p$  before container  $c'$ . In order to conserve the lifo principle, inequalities 2.2.12 and 2.2.13 are connected in

$$t^p(\mathcal{Z}_{cp}) + t^s(\mathcal{Z}_{cp}) \geq t^p(\mathcal{Z}_{c'p}) + t^s(\mathcal{Z}_{c'p}) + 1/L - L(2 - \alpha_{cc'p} - \delta_{cc'p}) \quad (2.2.16)$$

ensuring that container  $c$  is removed from position  $p$  after container  $c'$  if  $\alpha_{cc'p} = \delta_{cc'p} = 1$ .



### The complete CPP model

With the objective of positioning and reshuffling containers at lowest possible cost by minimizing the number of reshuffles and the total transportation time, the CPP model with the variable domains (2.2.17) can be stated as follows:

$$\text{minimize } x(\mathcal{C} \times \mathcal{N} \times \mathcal{P}) + t^m(\mathcal{C} \times \mathcal{N} \times \mathcal{I}) \quad (2.2.1)$$

subject to

$$t_{c10}^p = A_c, \quad \forall c \in \mathcal{C} \quad (2.2.2)$$

$$t^p(\mathcal{Z}_{ci} \mid n > 2, i = \mathcal{I}) = D_c, \quad \forall c \in \mathcal{C} \quad (2.2.3)$$

$$x(\mathcal{Z}_{cn}) \geq x(\mathcal{Z}_{cn+1}), \quad \forall c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1} \quad (2.2.4)$$

$$t_{cni}^m \geq T_{ij}(x_{cni} + x_{cn+1j} - 1), \quad \forall c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I \quad (2.2.5)$$

$$t_{cni}^m \leq T_{ij}(x_{cni} + x_{cn+1j} - 1) + L(2 - x_{cni} - x_{cn+1j}), \\ \forall c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I \quad (2.2.6)$$

$$t_{cni}^p + t_{cni}^s + t_{cni}^m \leq t_{cn+1j}^p + L(2 - x_{cni} - x_{cn+1j}), \\ \forall c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I \quad (2.2.7)$$

$$t_{cni}^p + t_{cni}^s + t_{cni}^m \geq t_{cn+1j}^p - L(2 - x_{cni} - x_{cn+1j}), \\ \forall c \in \mathcal{C}, n \in \mathcal{N}^{\rightarrow 1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I \quad (2.2.8)$$

$$t_z^p + t_z^s + t_z^m \leq 3Lx_z, \quad \forall z \in \mathcal{Z} \quad (2.2.9)$$

$$x(\mathcal{Z}_{cn}) \leq 1, \quad \forall c \in \mathcal{C}, n \in \mathcal{N} \quad (2.2.10)$$

$$x(\mathcal{Z}_{ci}) \leq 1, \quad \forall c \in \mathcal{C}, i \in \mathcal{I} \quad (2.2.11)$$

$$\alpha_{cc'p} \geq x(\mathcal{Z}_{cp}) + x(\mathcal{Z}_{c'p}) + [t^p(\mathcal{Z}_{c'p}) - t^p(\mathcal{Z}_{cp})]/L - 2, \\ \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.2.12)$$

$$\delta_{cc'p} \geq x(\mathcal{Z}_{cp}) + x(\mathcal{Z}_{c'p}) + [t^p(\mathcal{Z}_{cp}) + t^s(\mathcal{Z}_{cp}) - t^p(\mathcal{Z}_{c'p})]/L - 2, \\ \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.2.13)$$

$$\alpha_{cc'p} + \alpha_{c'cp} \leq 1, \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.2.14)$$

$$t^p(\mathcal{Z}_{cp}) + 1/L \leq t^p(\mathcal{Z}_{c'p}) + L(1 - \alpha_{cc'p}), \quad \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.2.15)$$

$$t^p(\mathcal{Z}_{cp}) + t^s(\mathcal{Z}_{cp}) \geq t^p(\mathcal{Z}_{c'p}) + t^s(\mathcal{Z}_{c'p}) + 1/L - L(2 - \alpha_{cc'p} - \delta_{cc'p}), \\ \forall (c, c') \in \mathcal{C}^2, p \in \mathcal{P} \quad (2.2.16)$$

$$x_z \in \mathbb{B}^{|\mathcal{Z}|}, (t_z^p, t_z^s, t_z^m) \in \mathbb{R}_+^{3|\mathcal{Z}|}; \quad (\alpha_{cc'i}, \delta_{cc'i}) \in \mathbb{B}^{2C(C-1)P} \quad (2.2.17)$$

### 2.2.2 Model analysis

The CPP model is polynomial with a order of  $\mathcal{O}(C^2P)$  variables and  $\mathcal{O}(C^2P)$  constraints. This is the same magnitude, relative to data, as the CTP model, only the scope of the problem considered is much smaller since each storage block is treated separately.

The design variables,  $x_z \in \{0, 1\}$ ,  $\forall z \in \mathcal{Z}$ , and the lifo variables,  $\alpha_{cc'p}, \delta_{cc'p} \in \{0, 1\}$ ,  $\forall (c, c') \in \mathcal{C}^2$ ,  $p \in \mathcal{P}$ , number  $CN(P+2) \simeq CNP$  and  $2C(C-1)P \simeq 2C^2P$  binary variables respectively and the time variables,  $t_z^p, t_z^s, t_z^m \in \mathcal{R}_+$ ,  $\forall z \in \mathcal{Z}$ , make  $3CN(P+2) \simeq 3CNP$  continuous variables. This results in approximately  $2C^2P+4CNP$  variables, with  $C^2P$  being the dominant term, assuming  $N \ll P < C$ .

The arrival and departure constraints make up  $2C$  equations, the flow restrictions number  $C(N-1) + 4C(N-1)(P+1)^2 + CN(P+2) + CN + C(P+2) \simeq 2CN + 4CNP^2 + CNP + CP$  constraints, and the lifo principle gives rise to  $5C(C-1)P \simeq 3C^2P$  inequalities. Again leaving out the negligible terms, this results in  $5C^2P + 4CNP^2 + CNP + CP + 2CN + 2C$  constraints altogether,  $C^2P$  being the dominant term.

The problem instance, presented in section 2.1.2, with a weekly throughput of 300 containers, 100 of them being assigned to a storage block with 20 positions, and 1-4 reshuffles per container, makes a problem instance with 432 thousand variables, 94.4 % of them binary and 92.6 % concerning the lifo restrictions, and approximately 1.65 million constraints, of which 1.0 million, corresponding to about 60 %, concern the lifo restrictions. Again, the lifo principle accounts for the vast majority of the variables and more than 50 % of the constraints.

Compared to the CTP model, the size reduction is significant since the number of variables and constraints is reduced by a factor 190 and 100 respectively. This is, however, partly due to the fact that the CPP model is formulated for each block separately whereas the CTP model concerns the entire terminal.

There are some limitations of the CPP model. No capacity restrictions are included which allows several containers to be moved simultaneously. This is a significant deviation from real-life situations where block cranes can carry at most two containers at a time, and usually only one.

Even though the CPP model is large and not solvable for real-life problem instances, the choice of the node variable  $x_{cni}$  reduces the size of the model significantly compared to the more traditional arc variable  $x_{cij}$ . Instead of representing container moves by an arc variable, doubling the position nodes  $(i, j)$ , the index  $n$  on the node variable  $x_{cni}$ , controlling the sequence of positions, gives rise to a much smaller number of variables as  $N \ll I$ . Though some constraints are simpler to formulate with the arc variable, the model size is generally reduced by using the node variable in the CPP model.

However, it would still be valuable to reduce the number of lifo variables and constraints and, furthermore, eliminating the “big  $M$ ” modeling technique, causing poor LP bounds when solving the model, would imply better performance.

### 2.3 A time-discretized BIP model for the CPP

The third modeling approach captures the same problem as the CPP model, presented in the previous section. The assumptions, discussed in sections 2.1 and 2.2, also applies to this model and the object as well as the problem restrictions are equivalent to the CPP model.

Aiming at reducing the number of variables and constraints concerning the lifo principle and achieving a formulation without a “big  $M$ ” notation, a time-discretized formulation is suggested: The CPPT model. The time dimension is discretized into time slots, enabling formulation of most of the restrictions in a very condensed way, including fewer types - and a smaller number - of variables, and resulting in a pure BIP model which is preferable to MIP models.

The CPPT model minimizes the total number of positionings and the total transportation time for containers, positioned in one storage block, subject to the three types of restrictions, also constituting the CTP and CPP models: Arrival/departure, flow, and lifo constraints, and - in addition to the two first models - the CPPT model includes a class of capacity constraints, observing the crane and stack capacity respectively. The model graph for each container is the same as for the CPP model, illustrated in figure 2.4 on page 44, only is the network repeated for each time slot in the discretized planning period.

The notation for the model is as follows. The sets,  $\mathcal{C}$ ,  $\mathcal{P}$ ,  $\mathcal{I}$ ,  $\mathcal{P}_0$ , and  $\mathcal{P}_I$ , are defined in the chapter introduction and section 2.2. In addition to these, the set  $\mathcal{T}$  contains all time slots in the planning horizon, discretized into  $t = t_1, t_1 + 1, \dots, T - 1, T$ , where  $t_1$  is the earliest arrival time  $\min\{A_c \mid c \in \mathcal{C}\}$  and  $T$  is latest departure time  $\max\{D_c \mid c \in \mathcal{C}\}$ . For modeling convenience, the arrival and departure times are shifted such that  $t_1 = 1$  and  $T = |\mathcal{T}|$ . For each container  $c$ , the time window  $\mathcal{T}_c = \{A_c, \dots, D_c\} \in \mathcal{T}$  is defined with  $T_c$  representing the average size of the sets  $|\mathcal{T}_c| = D_c - A_c$ .

Parameters in the CPPT model are the container arrival and departure times and the the node-to-node transportation times,  $A_c$ ,  $D_c$ , and  $T_{ij}$ , described in the chapter introduction, and the two capacity parameters,  $H$  and  $M$ , representing the the maximum stack height at each position in the block and moving capacity, i.e. the number of containers possible to carry by the crane(s) simultaneously. In contradistinction to the CTP and CPP models, the parameter  $L$ , representing the length of the planning horizon, is not included in the CPPT model which is due to elimination of “big  $M$ ” techniques, required in the first modeling approaches but rendered superfluous in this formulation.

The discretized time dimension enables us to introduce a new set of binary decision variables,  $x_{cti}$  and  $y_{ct}$ , substituting both the continuous time variables,  $t_{cni}^p$ ,  $t_{cni}^s$ , and  $t_{cni}^m$ , and the binary design variables,  $x_{cni}$ ,  $\alpha_{cc'i}$ , and  $\delta_{cc'i}$ , used in the CPP model. The new decision variables represent the position ( $x_{cti} = 1$ ) or moving ( $y_{ct} = 1$ ) of a container in a given time slot respectively. As opposed to the first two models, no lifo variables exist in the CTP model.

The technique for observing the lifo principle, illustrated in figure 2.6, is to detect situations where two containers,  $c$  and  $c'$ , are positioned at a position  $p$  in a time slot ( $t_k$

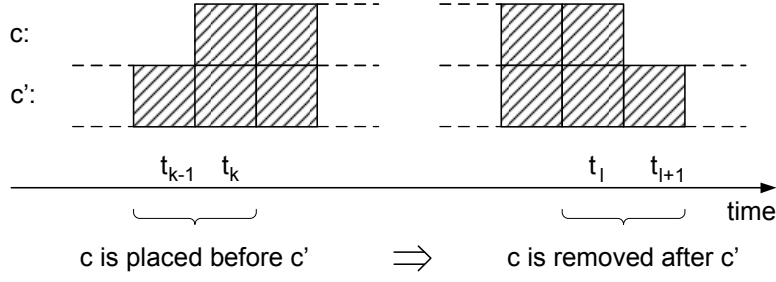


Figure 2.6: Observation of the lifo principle. If a situation with  $x_{ct_k p} = x_{c' t_k p} = 1$ ,  $x_{ct_{k-1} p} = 1$ , and  $x_{c' t_{k-1} p} = 0$ , indicating that container  $c$  is placed at position  $p$  before container  $c'$ , is detected, it must be ensured that  $x_{ct_l p} = x_{c' t_l p} = 1$ ,  $x_{ct_{l+1} p} = 1$ , and  $x_{c' t_{l+1} p} = 0$  for  $t_k < t_l$ .

in the figure), corresponding  $x_{ct_k p} = x_{c' t_k p} = 1$ , but only one of them in the previous time slot, corresponding to  $x_{ct_{k-1} p} = 1$ ,  $x_{c' t_{k-1} p} = 0$ , indicating that container  $c$  is placed first. Ensuring that in some later time slot ( $t_l$  in the figure) both containers are positioned at  $p$  and only the first arrival is remaining in the subsequent time slot observes the lifo principle.

The definition of the set  $\mathcal{Z}$  is changed to  $\mathcal{C} \times \mathcal{T} \times \mathcal{I}$ , i.e.  $z = (c, t, i) \in \mathcal{Z}$ , and the notation  $\mathcal{Q}^{k \rightarrow} = \mathcal{Q} \setminus \{q_1, q_2, \dots, q_k\}$  and  $\mathcal{Q}^{\rightarrow k} = \mathcal{Q} \setminus \{Q_n - k + 1, Q_n - k + 2, \dots, Q_n\}$ , introduced in section 2.2, is used in the CPPT model. Moreover, for notational convenience,  $\mathcal{Q}_q \cap \mathcal{Q}_{q'}$  is denoted  $\mathcal{Q}_{qq'}$  where  $\mathcal{Q}_{qq'}$  is the average size of the sets  $\mathcal{Q}_{qq'}$ .

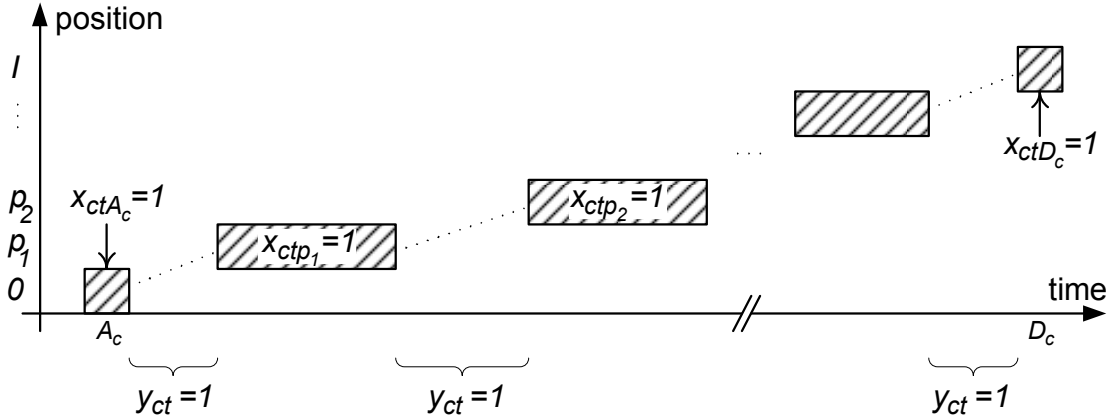


Figure 2.7: A sequence of positions for one container  $c$ . The hatched areas correspond to container  $c$  being positioned ( $x_{cti} = 1$ ) and the dashed lines between the positions correspond to container  $c$  being moved ( $y_{ct} = 1$ ).

An overview of the notation for the CPPT model can be found in table 2.3 and figure 2.7 provides a schematic overview of the solution structure of the CPPT.

|                                                       |                                                                              |
|-------------------------------------------------------|------------------------------------------------------------------------------|
| <b>Sets</b>                                           |                                                                              |
| $\mathcal{C} = \{1, \dots, C\}$                       | Containers                                                                   |
| $\mathcal{T} = \{1, \dots, T\}$                       | Set of time slots                                                            |
| $\mathcal{T}_c = \{A_c, \dots, D_c\} \in \mathcal{T}$ | Time window for container $c$                                                |
| $\mathcal{P} = \{1, \dots, P\}$                       | Positions                                                                    |
| $\mathcal{P}_0 = \{0, 1, \dots, P\}$                  | Positions $\mathcal{P}$ and arrival place $\{0\}$                            |
| $\mathcal{P}_I = \{1, \dots, P, I\}$                  | Positions $\mathcal{P}$ and departure place $\{I\}$                          |
| $\mathcal{I} = \{0, 1, \dots, P, I\}$                 | All nodes: Positions $\mathcal{P}$ , arrival and departure places $\{0, I\}$ |
| <b>Parameters</b>                                     |                                                                              |
| $A_c, c \in \mathcal{C}$                              | Arrival time for container $c$                                               |
| $D_c, c \in \mathcal{C}$                              | Departure time for container $c$                                             |
| $T_{ij}, i \in \mathcal{P}_0, j \in \mathcal{P}_I$    | Transportation time between nodes $i$ and $j$ , $i \neq j$                   |
| $H$                                                   | Maximum stack height                                                         |
| $M$                                                   | Crane capacity (maximum number of simultaneous moves)                        |
| <b>Decision variables</b>                             |                                                                              |
| $x_z \in \mathbb{B}^{CT_c I}$                         | 1 iff container $c$ is placed at position $i$ in time slot $t$               |
| $x_{cp}^m \in \mathbb{B}^{CP}$                        | 1 iff container $c$ is placed at position $p$ during storage                 |
| $y_{ct} \in \mathbb{B}^{CT_c}$                        | 1 iff container $c$ is moving in time slot $t$                               |

Table 2.3: Notation for the CPPT model.

### 2.3.1 The CPPT model

The CPPT model minimizes the total number of positionings plus the total transportation time in one storage block subject to 1) Arrival and departure constraints, 2) Flow constraints 3) Lifo constraints, and 4) Capacity constraints.

#### Objective

The objective function

$$x^m(\mathcal{C} \times \mathcal{P}) + y(\mathcal{C} \times \mathcal{T}_c) \quad (2.3.1)$$

sums the total number of positionings and the total transportation time for containers in the block. The CPPT model minimizes the objective function in order to reduce the cost of container positioning and reshuffling. Alternative objectives could be suggested but minimizing the handling time and the number of reshuffles is by far the most often required target in the business.

#### Arrival and departure constraints

The two equalities (2.3.2) and (2.3.3) ensure that the arrival and departure times are met by forcing each container to be at the arrival/departure place at the time of its arrival/departure respectively.

$$x_{cA_c0} = 1, \forall c \in \mathcal{C} \quad (2.3.2)$$

$$x_{cD_cI} = 1, \forall c \in \mathcal{C} \quad (2.3.3)$$

### Flow constraints

Constraints (2.3.4) - (2.3.7) concern flow restrictions, i.e. sequences of positions and movings of containers. In the equalities

$$x(\mathcal{Z}_{ct}) + y_{ct} = 1, \forall c \in \mathcal{C}, t \in \mathcal{T}_c \quad (2.3.4)$$

it is ensured that each container is at exactly one node ( $\sum_{i \in \mathcal{I}} x_{cti} = 1$ ) or moving between positions ( $y_{ct} = 1$ ) in all time slots between its arrival and departure times,  $A_c$  and  $D_c$ . In order to make sure that a container is not placed at a position more than once, i.e. that there are no “holes” in the sequence of time slots where a container is stored at a position, the inequalities

$$x_{ctp} \leq 1 - x_{ct'p} + x_{ct'-1p}, \forall c \in \mathcal{C}, t \in \mathcal{T}_c^{-1}, t' \in \mathcal{T}_c^{1 \rightarrow}, t < t', p \in \mathcal{P} \quad (2.3.5)$$

detect situations with  $x_{ct'p} = 1$  and  $x_{ct'-1p} = 0$ , i.e. where container  $c$  is placed at position  $p$  in time slot  $t'$ , and set all values of  $x_{ctp} = 0$  for  $t < t'$ . Two inequalities observe the moving restrictions. First, it is ensured that if container  $c$  leaves node  $i$  at time slot  $t$ , it will be at the subsequent node  $j$  exactly  $T_{ij}$  later. Next, for all time slots  $t$  between the last at node  $i$  and the first at node  $j$ , the moving variables  $y_{ct}$  must equal 1. For notational convenience, these conditions are formulated using standard algebraic notation as follows.

$$\sum_{j \neq i} x_{ct+T_{ij}j} \geq x_{ct-1i} - x_{cti}, \forall c \in \mathcal{C}, t \in \mathcal{T}^{1 \rightarrow}, i \in \mathcal{P}_0 \quad (2.3.6)$$

$$\sum_{t < t' < t+T_{ij}+1} y_{ct'} \geq T_{ij}(x_{cti} + x_{ct+T_{ij}+1j} - 1), \forall c \in \mathcal{C}, t \in \mathcal{T}^{-1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I \quad (2.3.7)$$

### Lifo constraints

Constraints (2.3.8) - (2.3.10) concern situations where two containers share the same position. Recalling that  $\mathcal{T}_{cc'} = \mathcal{T}_c \cap \mathcal{T}_{c'}$ , clearly the smallest and largest elements in  $\mathcal{T}_{cc'}$  is  $\max\{A_c, A_{c'}\}$  and  $\min\{D_c, D_{c'}\}$  respectively. The inequalities

$$x_{ctp} + x_{c'tp} \leq 1 + x_{ct-1p} + x_{c't-1p}, \forall (c, c') \in \mathcal{C}^2, t \in \mathcal{T}_{cc'}^{1 \rightarrow}, p \in \mathcal{P} \quad (2.3.8)$$

$$x_{ctp} + x_{c'tp} \leq 1 + x_{ct+1p} + x_{c't+1p}, \forall (c, c') \in \mathcal{C}^2, t \in \mathcal{T}_{cc'}^{-1}, p \in \mathcal{P} \quad (2.3.9)$$

ensure that two containers,  $c$  and  $c'$ , is not placed at or removed from a position  $p$  in the same time slot. This is a necessary condition, both to match the actual operations in the container yard where the handling equipment can only place one container at a position at a time, and in order to ensure the lifo principle, formulated in constraints (2.3.10), executed for all pairs of containers  $(c, c') \in \mathcal{C}^2$ , time slots  $t \in \mathcal{T}_{cc'}^{2 \rightarrow}$  and  $t' \in \mathcal{T}_{cc'}^{1 \leftarrow 1}$  with

$t > t'$ , and positions  $p \in \mathcal{P}$ . As well as in constraints (2.3.5), a reference time slot  $t'$  is used to preserve the lifo conditions in inequalities

$$x_{ctp} \geq x_{c'tp} - (2 - x_{ct'p} - x_{c't'p} + x_{c't'-1p}) \quad (2.3.10)$$

detecting situations with  $x_{ct'p} = 1$ ,  $x_{c't'p} = 1$ , and  $x_{c't'-1p} = 0$ , i.e. where container  $c$  is placed at position  $p$  before container  $c'$ , positioned in time slot  $t'$ , thus enforcing all  $x_{ctp} \geq x_{c'tp}$  for  $t > t'$ .

### Capacity constraints

Constraints (2.3.11) and (2.3.12) conserve the capacity restrictions by

$$y(\mathcal{C}) \leq M, \forall t \in \mathcal{T} \quad (2.3.11)$$

$$x(\mathcal{Z}_{tp}) \leq H, \forall t \in \mathcal{T}, p \in \mathcal{P} \quad (2.3.12)$$

ensuring that no more containers than the crane capacity allows are moved at the same time and that the maximum stack height is not violated.

### The complete CPPT model

Seeking the optimal sequences of positions for each container by minimizing the total number of reshuffles and the total transportation time, observing the problem restrictions together with the variable domains (2.3.13), the CPPT model can be stated as:

$$\text{minimize } x^m(\mathcal{C} \times \mathcal{P}) + y(\mathcal{C} \times \mathcal{T}_c) \quad (2.3.1)$$

subject to

$$x_{cA_0} = 1, \forall c \in \mathcal{C} \quad (2.3.2)$$

$$x_{cD_cI} = 1, \forall c \in \mathcal{C} \quad (2.3.3)$$

$$x(\mathcal{Z}_{ct}) + y_{ct} = 1, \forall c \in \mathcal{C}, t \in \mathcal{T}_c \quad (2.3.4)$$

$$x_{ctp} \leq 1 - x_{ct'p} + x_{ct'-1p}, \forall c \in \mathcal{C}, t \in \mathcal{T}_c^{-1}, t' \in \mathcal{T}_c^{1 \rightarrow}, t < t', p \in \mathcal{P} \quad (2.3.5)$$

$$\sum_{j \neq i} x_{ct+T_{ij}j} \geq x_{ct-1i} - x_{cti}, \forall c \in \mathcal{C}, t \in \mathcal{T}^{1 \rightarrow}, i \in \mathcal{P}_0 \quad (2.3.6)$$

$$\sum_{t < t' < t+T_{ij}+1} y_{ct'} \geq T_{ij}(x_{cti} + x_{ct+T_{ij}+1j} - 1), \forall c \in \mathcal{C}, t \in \mathcal{T}^{-1}, i \in \mathcal{P}_0, j \in \mathcal{P}_I \quad (2.3.7)$$

$$x_{ctp} + x_{c'tp} \leq 1 + x_{ct-1p} + x_{c't-1p}, \forall (c, c') \in \mathcal{C}^2, t \in \mathcal{T}_{cc'}^{1 \rightarrow}, p \in \mathcal{P} \quad (2.3.8)$$

$$x_{ctp} + x_{c'tp} \leq 1 + x_{ct+1p} + x_{c't+1p}, \forall (c, c') \in \mathcal{C}^2, t \in \mathcal{T}_{cc'}^{-1}, p \in \mathcal{P} \quad (2.3.9)$$

$$x_{ctp} \geq x_{c'tp} - (2 - x_{ct'p} - x_{c't'p} + x_{c't'-1p}), \\ \forall (c, c') \in \mathcal{C}^2, t \in \mathcal{T}_{cc'}^{2 \rightarrow}, t' \in \mathcal{T}_{cc'}^{1 \leftrightarrow 1}, t > t', p \in \mathcal{P} \quad (2.3.10)$$

$$y(\mathcal{C}) \leq M, \forall t \in \mathcal{T} \quad (2.3.11)$$

$$x(\mathcal{Z}_{tp}) \leq H, \forall t \in \mathcal{T}, p \in \mathcal{P} \quad (2.3.12)$$

$$x_{cti} \in \{0, 1\}, \forall c \in \mathcal{C}, t \in \mathcal{T}_c, i \in \mathcal{I}; \quad y_{ct} \in \{0, 1\}, \forall c \in \mathcal{C}, t \in \mathcal{T}_c \quad (2.3.13)$$

### 2.3.2 Model analysis

Just as the two previous models the CPPT model is polynomial. It has a order of  $\mathcal{O}(CT_cP)$  variables and  $\mathcal{O}(C^2T_{cc'}^2P)$  constraints. The largest set in both the CTP and the CPP model is  $\mathcal{C}$  representing the containers in the system. In the CPPT model, however, the set  $\mathcal{T}$  constitute the largest number of elements and  $\mathcal{P}$  is the smallest set in the model.

By substituting the previously used decision variables,  $x_{cni}$ ,  $t_{cni}^p$ ,  $t_{cni}^s$ ,  $t_{cni}^m$ ,  $\alpha_{cc'i}$ , and  $\delta_{cc'i}$ , with the binary  $x_{cti}$  and  $y_{ct}$ , the number of variables is significantly reduced compared to the CTP and the CPP model. On the other hand, the number of constraints is larger than in the two first models.

The position variables  $x_{cti} \in \{0, 1\}$ ,  $\forall c \in \mathcal{C}, t \in \mathcal{T}_c, i \in \mathcal{I}$ ,  $x_{cp}^m \in \{0, 1\}$ ,  $\forall c \in \mathcal{C}, p \in \mathcal{P}$  and the moving variables  $y_{ct} \in \{0, 1\}$ ,  $\forall c \in \mathcal{C}, t \in \mathcal{T}_c$  number  $CT_cI = CT_c(P+2) \simeq CT_cP$  and  $CT_c$  binary variables respectively, resulting in a total number of  $CT_cP + CP + CT_c$  variables.

The constraints are made of  $2C$  arrival and departure equations,  $CT_c + C(T_c - 1)^2P + 2C(T - 1)(P + 1) \simeq CT_c + CT_c^2P + 2CTP$  flow restrictions,  $2C(C - 1)(T_{cc'} - 1)P + C(C - 1)(T_{cc'} - 2)^2P \simeq 2C^2T_{cc'}P + C^2T_{cc'}^2P$  inequalities concerning the lifo principle, and  $T + TP$  capacity restrictions, resulting in an upper bound of  $C^2T_{cc'}^2P + 2CTP + CT_c^2P + 2C^2T_{cc'}P + CT_c + TP + T + 2C$ , with  $C^2T_{cc'}^2P$  being the dominant term.

As well as for the analysis of the CTP and CPP models in sections 2.1.2 and 2.2.2, the size of the CPPT model is illustrated by the previously described problem instance with 300 containers, 100 of them assigned to one of the 10 storage blocks with 20 positions. With a planning horizon of one week, discretized into 336 time slots (corresponding to a discretization level of 10 minutes through the 7 days of 8 working hours) the number of constraints amounts to approximately 350 thousand variables, all binary, and 1.5 billion constraints, 96 % concerning the lifo principle.

This corresponds to a reduction of the number of variables by a factor 230 compared to the CTP model and a decrease of about 20 % compared to the CPP model. However, the number of constraints increases significantly - by a factor of 10 and 1000 respectively - which is due to the flow and lifo restrictions, especially (2.3.5), preserving the continuity, and (2.3.10), observing the lifo principle, all executed for combinations of two time slots which implies a large number of constraints.

Although a large number of constraints in general increases the solution time, this also restricts the problem and tightens the formulation. Together with the reduced number of variables, also contributing to a much smaller solution space, this leads us to believe in possibly shorter running times.

Besides being more true to reality by including capacity restrictions, the primary gains of the CPPT model are the elimination of the “big  $M$ ” technique, the MIP model structure, and the reduction of the number of variables. It is well-known that “big  $M$ ” terms imply a poor linear programming relaxation and low-quality bounds when solving models with a branch and bound algorithm, as is the case in standard solvers, and that MIP models are, in general, less tractable than pure BIP models. Altogether, this indicates a possible advantage of the CPPT model compared to the first two models.



### Further research ideas for the CPPT model

A potentially promising perspective for improving the CPPT model and achieving shorter running times is reducing the number of time slots included in the model. The largest set in the CPPT model  $\mathcal{T}$  contains all consecutive time slots in the discretized planning period. Reducing the number of time slots to be “active” in the model would imply a much smaller model and, thereby, improve its performance.

Realizing that containers are only moved in time slots around the different arrival and departure times and that no reshuffling takes place in much of the intervening time, clearly only a subset of the time slots is required to assign new positions to reshuffled containers. To represent this property, a sparse set of time slots  $\mathcal{T}^s \subseteq \mathcal{T}$ , consisting of “key points in time” where reshuffling may take place, can be introduced.

These key time slots are clusters, centered around the arrival and departure times,  $A_c$  and  $D_c$ , constituting the backbone of the sparse set  $\mathcal{T}^s$ . The number of consecutive time slots to be included around each arrival and departure time depends on the size of the block and the transportation times between positions. It must be ensured that there is sufficient time to move containers from any position  $i \in \mathcal{P}_0$  to any other position  $j \in \mathcal{P}_I$  and that containers placed on top of a container to be reshuffled can be moved within the respective time slots. Therefore, the number of time slots in each cluster represents the worst-case time required to reposition all containers in a stack, i.e.  $2k+1 \geq H \max\{T_{ij} \mid i \in \mathcal{P}_0, j \in \mathcal{P}_I\}$ , where  $k$  is the number of consecutive time slots to be included before and after each arrival and departure time, except for before the smallest and after largest element in the set  $\{t_1, T\}$ . Lower values of  $k$  would imply a smaller set  $\mathcal{T}^s$ , but in return optimal solutions could be cut off.

Consequently,  $\mathcal{T}^s$  consists of the union of  $2C$  clusters of time slots of size  $2k+1$  with the arrival and departure times as center elements:  $\mathcal{T} \supseteq \mathcal{T}^s = \{A_1, A_1+1, \dots, A_1+k\} \cup \{A_2-k, \dots, A_2+k\} \cup \dots \cup \{D_{C-1}-k, \dots, D_{C-1}+k\} \cup \{T-k, \dots, T\}$ . Figure 2.8 shows the principle of two sets of time slots,  $\mathcal{T}$  and  $\mathcal{T}^s$ .

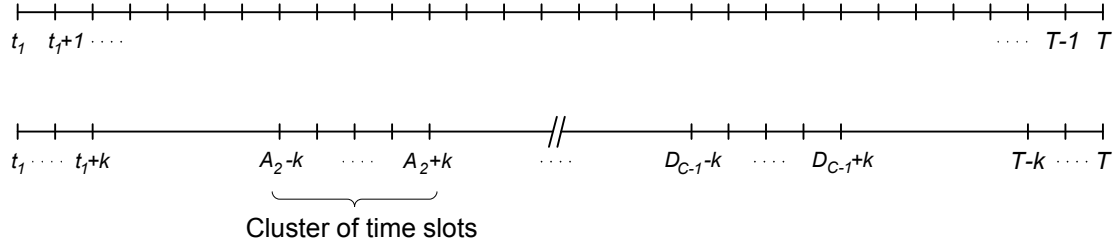


Figure 2.8: Principal overview of the two sets of time slots: All consecutive time slots  $\mathcal{T}$  and the sparse set of time slots  $\mathcal{T}^s$ , consisting of clusters of  $2k+1$  time slots centered around the arrival and departure times.

Depending on the ratio of the number of time slots and containers, the gain of introducing a sparse set  $\mathcal{T}^s$  instead of including the entire time dimension  $\mathcal{T}$  throughout

the whole model may be quite considerable. However, some modeling challenges must be overcome, especially concerning the moving restrictions if allowing moves - but not positionings - between key time slots.

Making use of a sparse set  $\mathcal{T}^s$  represents an event-based approach which can be useful - not only for modeling purposes - but also for developing solution methods. Chapter 3 describes a heuristic approach to the CPP, building on the event-based principles discussed in this section.

## 2.4 Conclusive remarks

For comparison of the CTP, the CPP, and the CPPT model, table 5.76 provides an overview of the order of the three models together with the number of binary and continuous variables, and the arrival & departure, flow, lifo, and capacity constraints respectively. Furthermore, the model sizes based on the problem instance, described in sections 2.1.2, 2.2.2, and 2.3.2, are stated.

|                    | 1) CTP model                        | 2) CPP model              | 3) CPPT model                  |
|--------------------|-------------------------------------|---------------------------|--------------------------------|
| <b>Order</b>       |                                     |                           |                                |
| <i>Var.</i>        | $\mathcal{O}(C^2P)$                 | $\mathcal{O}(C^2P)$       | $\mathcal{O}(CT_cP)$           |
| <i>Con.</i>        | $\mathcal{O}(C^2P)$                 | $\mathcal{O}(C^2P)$       | $\mathcal{O}(C^2T_{cc'}^2P)$   |
| <b>Variables</b>   |                                     |                           |                                |
| <i>Bin.</i>        | $C(S+P)(P+V) + 3C^2P$               | $CNP + 2C^2P$             | $CT_cP + CP + CT_c$            |
| <i>Cont.</i>       | $C(S+P+V)$                          | $3CNP$                    |                                |
| <i>Lifo</i>        | $3C^2P$                             | $2C^2P$                   |                                |
| <b># tot</b>       | $82 \cdot 10^6$                     | $432 \cdot 10^3$          | $350 \cdot 10^3$               |
| <b>Constraints</b> |                                     |                           |                                |
| <i>A&amp;D</i>     | $2C + \sum_{k=1}^{C-1} (C-k) + 2CV$ | $2C$                      | $2C$                           |
| <i>Flow</i>        | $2CP + C(S+P)(P+V)$                 | $4CNP^2 + CNP + 2CN + CP$ | $CT_c + CT_c^2P + 2CTP$        |
| <i>Lifo</i>        | $8C^2P$                             | $5C^2P$                   | $2C^2T_{cc'}P + C^2T_{cc'}^2P$ |
| <i>Cap.</i>        |                                     |                           | $T + TP$                       |
| <b># tot</b>       | $172 \cdot 10^6$                    | $1.65 \cdot 10^6$         | $1.5 \cdot 10^9$               |

Table 2.4: Comparison of the CTP, CPP, and CPPT model. The first part states the order of the models with respect to variables and constraints. Next, the number of binary, continuous, and lifo variables and the number of constraints of the different types are shown together with the sizes of the problem instances, described in the three model analysis sections.

The advances in the modeling process concern partly reductions in model sizes - primarily in the number of variables - and partly improvements in model structure. The number of variables is reduced by a factor 190 from the CTP to the CPP model and another 20 % reduction is achieved in the CPPT model, resulting in a total reduction of a factor 230 from the first to the third model. The number of constraints is reduced by a factor 100 in the CPP model but is increased in the CPPT model with a factor of 10 compared to the CTP model. As regards the lifo principle, the number of variables

dedicated to observing lifo restrictions is reduced from  $54 \cdot 10^6$  (66 %) in the CTP model to  $400 \cdot 10^3$  in the CPP model and no lifo variables exist in the CPPT model. The number of lifo constraints varies from  $144 \cdot 10^6$  in the CTP model over 1 million in the CPP model to about  $1.4 \cdot 10^9$  in the CPPT model. Also the model structure differs in the three models. The first two are MIP models with 99.8 % and 94.4 % binary variables respectively. The CPPT model, on the other hand, is a pure BIP model, a property that is well-known to improve solution performance when applying branch and bound algorithms.

There are several perspectives for improving solution performance of the models. Firstly, a number of the variables may be redundant, especially the lifo variables in the CTP and CPP models as many of the containers will not conflict in practice. Correspondingly, the large number of lifo constraints indicate that it may be advantageous to relax these and add them on the fly. The large number of constraints - as opposed to the number of variables - in the CPPT model suggests that this approach may imply good results. Furthermore, alternative solution techniques such as Lagrangian Relaxation or other decomposition approaches may have advantages. If relaxing families of complicating constraints enables fast solution of the remaining subproblem, finding optimal solutions within reasonable time may be obtainable.

Concluding, the proposed models will, due to their size in variables and constraints, scale in an undesired way and, therefore, in all probability not being able to constitute an efficient basis for solving large-scale problems, especially not if using standard optimizers. Furthermore, the structure of the problem and the very flat-curved objective function imply a large set of equally good solutions. This leads to a high degree of symmetry, as many paths through the position network of the block can be mirrored or shifted without affecting the objective function. For such problems where a large part of the decision variables to a great extent can be chosen arbitrarily among a large number of possibilities, it may be difficult to close the optimality gap.

The models represent the process and results of thorough work on refining mathematical representations of the CPP and constructing a good compact formulation of the problem. However, being complex - especially due to the lifo restrictions - and not resembling other problems for which well-known solution techniques exist, the problem is not easily represented in a compact way with good scalability in increasing problem size. Whether it is possible to find such a compact model for the CPP is, by the end of this research study, uncertain - but assumed unlikely. It may, however, be interesting to investigate in future projects. The present work indicates that focus should then be on representing the lifo restrictions and the time dimension in an efficient way and, moreover, that it may be advantageous to invest time in transforming well-known problems, such as VRP, to a subproblem for which additional required constraints may be formulated, possibly enabling efficient use of decomposition approaches.

For the VRP, efficient decomposition methods exist. One way to reduce the solution space significantly is to make use of a path formulation and iteratively generate and add new paths to the solution process. As discussed, a straightforward idea is to transform the CPP into the VRP, letting containers be the vehicles to visit positions rather than

customers. However, a fundamental difference between the CPP and the VRP is that such paths for containers are not disjunct but, rather, weaved together by the customers, alias the positions, being placed on top of each other. The side constraints, required to preserve the problem structure, include the lifo restrictions which are not easily formulated. Therefore, even though the idea of such a problem transformation and relaxation or decomposition approach may seem plausible, it is believed that this research direction may be intractable [50].

Following the next chapter, presenting a heuristic algorithm for the CPP, different approaches to solving the CPP and CPPT models are reported in chapter 5. Capturing the entire port terminal, the CTP model is not well-suited for solving larger problem instances and, therefore, is not considered in the remaining thesis. The CPP and CPPT models, however, are validated and tested by a large number of problem instances, tests with a large run time time limit are performed, and an investigation of the potential gain in computation time when relaxing the lifo constraints is carried out.



## Chapter 3

# A new event-based heuristic for the CPP

In this chapter a heuristic for the container positioning problem (CPP) is presented. The models presented in chapter 2 are extensive and complex in the representation of particularly the life restrictions. The models each have a very large number of variables and constraints and although, in general, it is not correct to claim that computation time increases and computational feasibility decreases as the number of variables and constraints increases, it is reasonable to expect that solving these models is very hard.

Later, in chapter 5 on the computational results, it will in fact become clear that the lower bounds, defined by the linear programming relaxation of the models, are not very sharp or tight. Most standard integer programming solvers are very dependent on the sharpness of the bounds which are used in the branch and bound search. In the case of the CPP models, there is a large gap between the root node values and the optimal solutions, resulting in extensive branching in order to close the gap even for quite small test problems. As described in chapter 4, real-life CPP instances give rise to very large data sets which limits the possibility of using the formulations to solve the CPP to optimality.

The goal of this PhD study is to solve real-life instances of the CPP and although an optimization-based method to solving the problem is not ruled out, in order to achieve this goal, the focus in this second part of the thesis is on heuristics.

The heuristic is developed for the CPP, i.e. it considers the containers from they arrive at their storage block till they depart from it again. The assumptions described in the presentation of the three mathematical models in sections 2.1, 2.2, and 2.3 also apply to the CPP heuristic. The approach is a greedy construction algorithm with no randomization and is, by definition, deterministic.

In general, one distinguishes between two types of heuristics - or techniques used in heuristics: Construction and improvement procedures. The first approach applies some predefined rules and builds the solution from scratch whereas the latter is based on an initial solution and makes some qualified changes in order to achieve better ones [62, 39]. Note that there is a gray area between construction heuristics and simulation approaches

as the principle of following a set of rules to build the solution is applied in both cases.

The intuition of the CPP heuristic is to handle decisions in an on-line fashion. In practice, when a container is to be positioned, e.g. after its arrival, the decision must be taken quickly and based on the current status of the block. Information about future containers to arrive or operations to be performed is often uncertain due to changes in plans for the trucking companies, administrative issues concerning customs or safety, etc. Therefore, constructing the solution “container by container” is believed to be the best suited approach and, furthermore, is a very intuitive way to handle the CPP.

Typically consuming very little time, a greedy construction algorithm has some significant advantages. Decisions can be made very quickly, possibly leaving time to evaluate and/or improve the given solution e.g. by performing re-runs with different parameter settings, applying repair functions, etc. Furthermore, such algorithms are usually not sensitive to increases in problem sizes for which reason it may be very suitable for practical application.

Clearly, the advantage of developing a heuristic for a problem is not having to examine all possible solutions but only a number of promising ones. The way this is done in this study is to apply an event-based approach where only key time slots are considered. The rough idea behind this short cut through the system was touched on in section 2.3.2, presenting the potentials of considering only a number of key time slots instead of including the entire discretized time horizon in the CPPT model.

Building on an outer loop over a number of key events the heuristic draws on intelligent and promising principles and rules when evaluating and selecting positions, at each event aiming at minimizing the expected number of reshuffles and transportation times at subsequent events.

Figure 3.1 provides an overview of the main components of the heuristic. The algorithm consists of two steps: Initialization and event processing. Three main functions handle the events: 1 positionIncomingContainer, 2 repositionToBufferZone, and 3 removeOutgoingContainer, depending on the type of event. Within these, five sub-functions are applied: A evaluateCandidatePositions, B selectMostFavourablePosition, C removeFromCurrentPosition, D moveToNewPosition, and E repositionTopStackContainers. The four functions, A - D, are used when positioning or reshuffling containers. If containers are placed on top of the one to be repositioned, which cannot occur with incoming containers, function E moves these before functions A - D handle the actual container. Furthermore, one supplementary function is called by E when reshuffling stack containers: F repositionCloseToCurrentPosition, not represented in the figure. Within F, functions A - D are again used to reposition each stack container, hindering access to the actual one. Depending on the type of positioning - placing an incoming container, reshuffling a stack container on account of a lower placed one, or moving a container close to its departure time - different actions are taken if no feasible position is found by functions A and B. An incoming container which is not possible to place at one of the positions near to the arrival place is sought repositioned in the buffer zone (a specified number of positions at the departure end of the block), equivalently, a stack container which cannot be placed near to the current position is sought repositioned in the buffer

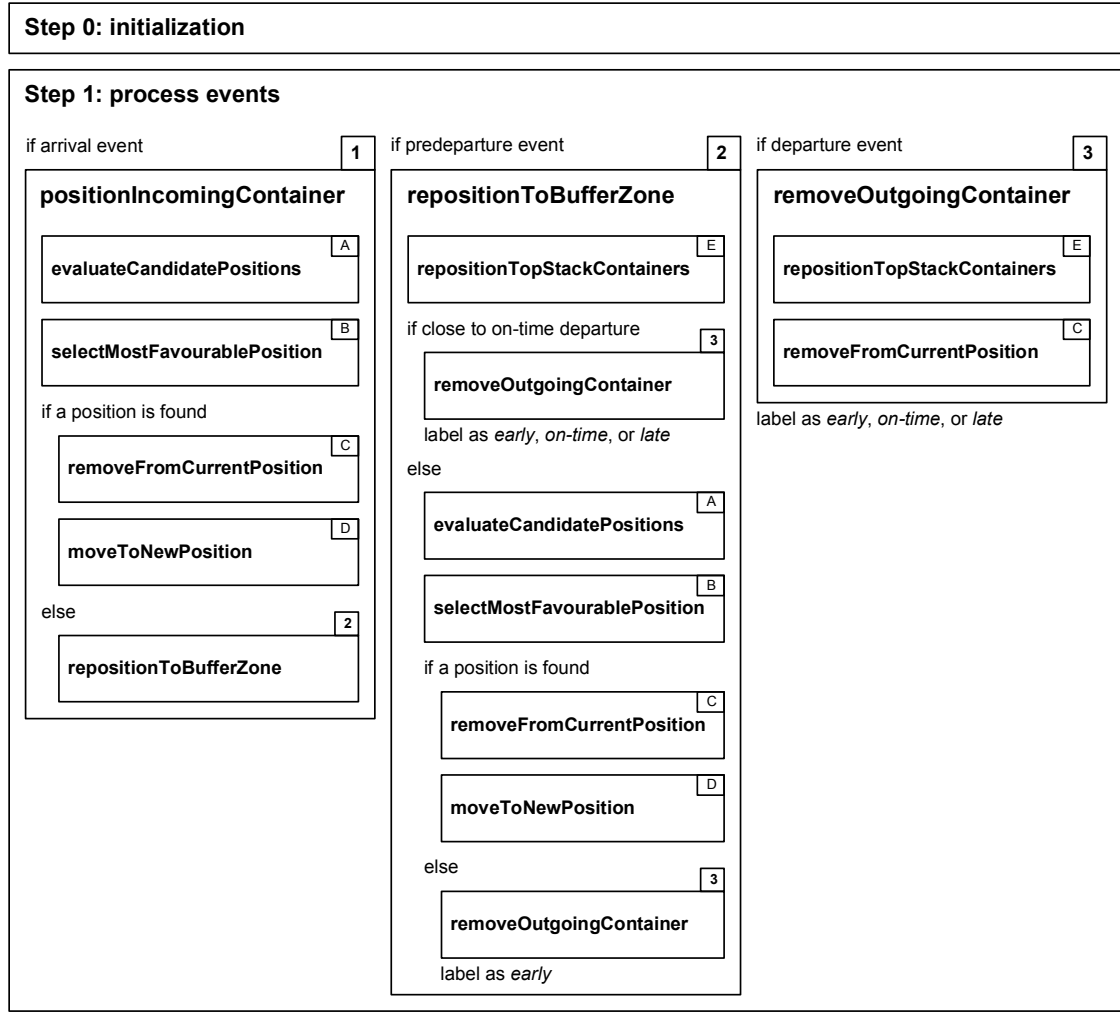


Figure 3.1: Overview of the main components of the heuristic. Step 0 initializes parameters and Step 1 represents the main loop. The three main functions are 1, 2, and 3, and the five subfunctions are A, B, C, D, and E. The supplementary function F, called by E, is not represented in the figure.

zone, and a container close to its departure times for which there is no feasible position in the buffer zone is moved to the departure place.

In the following section the principal concepts and elements in the heuristic are described. Subsequently, a pseudo code for the algorithm is presented, attended with thorough descriptions of each step, function, and procedure. Finally, a discussion of the approach concludes the chapter.



## 3.1 Concepts

The backbone of the algorithm is the list of events, consisting of points in time where certain operations should be started. Each event represents either arrival, predeparture (corresponding to reshuffling to prepare for a departure), or departure operations. When positioning containers - in case of an arrival or a predeparture event - two main procedures of the algorithm come into play: Evaluation of candidate positions and selection of one. In the evaluation procedure candidate positions are categorized according to four categorization criteria, evaluated by three evaluation parameters, and finally, a quality function is applied, resulting in a quality stamp for each candidate position by which they are sorted. In the selection procedure the first position in the list of quality values to meet a number of feasibility criteria which ensure legal positionings with no future time conflicts is chosen as the new position for the container to be placed.

The following description of the list of events and the procedures executed for the three types of events corresponds to the event processing step and the three main functions, 1 positionIncomingContainer, 2 repositionToBufferZone, and 3 removeOutgoingContainer. The subsequent descriptions of the evaluation and selection procedures in connection with positioning of containers correspond to the two central subfunctions, A evaluateCandidatePositions and B selectMostFavourablePosition. First, the concept system time is described.

### 3.1.1 System time

Several procedures in the heuristic make use of time-dependent operations. To keep track of time consumption when moving containers, preparing for future moves, etc., a system time  $T^S$  is introduced. The system time corresponds to a “clock” on the crane, continuously updated when moves have been performed and when event are processed. Thus,  $T^S$ , always corresponding to the actual time, is used when processing the list of events, when checking feasibility of candidate positions, and when performing moves between positions in the block.

### 3.1.2 List of events

The idea behind the list of events is to efficiently control the time dimension and ensure execution of the required operations. As mentioned previously, three types of operations can be identified: Positioning in the block after arrival, reshuffling between block positions, and moving to the departure place to meet the departure time. Reshuffling of a container may occur on account of itself or due to another container in the stack and several reshuffles may be required for some containers. All such operations - corresponding to the sequences of moves for each container in the system - are to be performed on a common time axis in the best possible way and for that purpose the list consisting of three types of events - arrival, predeparture, and departure - is introduced.

Knowing the arrival and departure times for all containers to be stored in the block, the respective points in time can be arranged in a list and sorted by descending time

values. Initially, only arrival and departure times are needed - as well as known - but when starting to position containers in the block, intermediate points in time where moves must take place in order to meet departure times are required and, later, changes in the stacks may necessitate additional time stamps. These intermediate points in time are referred to as predeparture times and their respective events constitute a tool for handling dynamics in the system since they are continuously updated whereas arrival and departure times are constant.

Thus, each event in the list corresponds to a specific container and a point in time where either a moving operation (occasioned by an arrival or a reshuffle) should be started or a container should be removed from the block (in order to meet its departure time). The time it takes to perform the required moves influences the list of events since containers' predeparture times may be changed if they are reshuffled.

Section 2.3.2, presenting the ideas for reducing the solution space for the CPPT model, suggested a set of key time slots as a number of clusters, centered around the arrival and departure times. In the CPP heuristic, the key time slots, corresponding to the events, are a lot fewer, initially only representing arrival and departure times. During the time containers are stored in the block reshuffling may be needed, thus necessitating one or more intermediate time slots, as well as points in time prior to the departure times - sufficiently early to move containers to the departure place - must be included. Contrary to the CPPT model where a number of time slots around each arrival and departure time must be included throughout the entire solution procedure to ensure that potentially required moves are possible to carry out, in the heuristic, the exact time to start such a moving operation can be determined when knowing the container's current position. Therefore, the additional time slots can be adjusted dynamically so that unnecessary points in time can be avoided.

### **Initializing and updating the list of events**

The list of events is initialized such that there is an arrival, a predeparture, and a departure event for each container. The respective time stamps equal the arrival times in the first case whereas both the predeparture and departure events are assigned the departure time. The predeparture events are later updated according to the time needed to ensure that positioned containers meet their departure times. Thus, the length of the list of events throughout the algorithm is three times the number of containers. However, the number of operations concerning each container may be more than three due to reshuffles, implying changes in the predeparture times for all containers that are affected by the respective event operations. After initialization the list is sorted by ascending time stamps. Figure 3.2 illustrates the principle of the initial list of events before the sorting.

While arrival and departure events remain unchanged throughout the algorithm, predeparture events may be altered several times. The first time a predeparture event is updated is when the respective container is positioned in the block after arrival. Knowing the time required to move the container from the position to the departure place, the time stamp of the predeparture event is adjusted by subtracting the transportation time

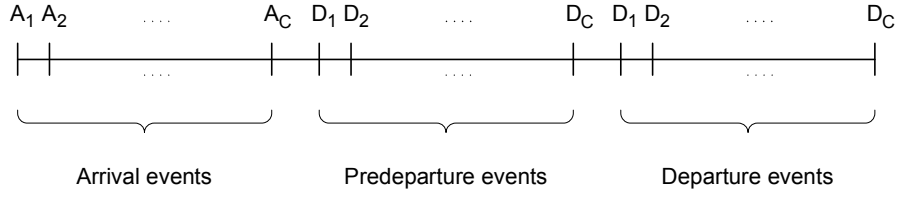


Figure 3.2: Schematic overview of the list of events when initialized. For each container, an arrival, a predeparture, and a departure event is initialized, time stamps equal to the arrival and departure times,  $A_c$  and  $D_c$ . During the solution procedure, predeparture events are updated and the list of events is continuously sorted by ascending time values.

between the position and the departure place from its departure time. If at a later point in time, a container is placed on top of it, the predeparture time is updated again, now subtracting the time it would take to reshuffle the upper placed container before moving to the departure place. Consequently, every time a container is placed in a stack, the predeparture events for all containers in the stack are updated.

Though no new predeparture events are added, each container may be reshuffled several times, occasioned by other containers' predeparture or departure events since all containers placed on top of a container to be removed are repositioned and assigned new predeparture times. Thus, when containers are reshuffled on account of an other container, new predeparture times - possibly earlier or later than the previous ones - are introduced, resulting in a larger total number of moves than events.

### Procedures for each event

The procedure for each event is to carry out a sequence of operations based on the type of event: Arrival, predeparture, or departure. In case of an arrival event, the incoming container is sought positioned close to the arrival place. The search for the best possible position is carried out within a specified range, counting from the first available position in the block. If no feasible position is found the container is placed at the farthest end of the block within a number of positions - also limited by the specified range - called the buffer zone, counting the last available positions in the block.

In case of a predeparture event, any containers that may be placed on top of the actual one are removed from the stack in order to access the container related to the event. The containers to be repositioned first are sought placed close to the current position. As for positioning of incoming containers, the search for the best possible position for each stack container is carried out within the specified range, now either counting half the range backwards and half the range forwards from the current position or counting the full range forwards, depending on the chosen strategy. If no feasible position is found for a stack container, it is placed at the buffer zone. After reshuffling the stack containers the actual container can be handled. Being close to its departure time the container may best be moved directly to the departure place if it results in an

on-time or close to on-time departure. In that case, the container is removed from the block and labeled either early, on-time, or late departed. Otherwise, the best possible position is searched for within the buffer zone. If no feasible position is found here, the container is removed from the block and labeled early departed.

In case of a departure event, if not already removed from the block during its predeparture event, the container is moved from its stack in the buffer zone to the departure place. If any containers are placed on top of it, they are reshuffled to a position nearby, using the same routines as described above.

When processing arrival or predeparture events, i.e. when positioning an incoming container or reshuffling a stacked one, a certain number of routines interact. As briefly described in the chapter introduction, four functions handle the main processes in the heuristic: Evaluation of candidate positions, selection of the best one feasible, and moving the container to the new position. These routines are repeated every time a container - just arrived, to be moved towards the departure place, or to be reshuffled on account of an other one - is to be positioned. When one or several moves have been performed, the respective predeparture events for all containers affected by the operations are updated to the point in time where they must be moved to the departure place in order to meet their departure times.

### **Possibly overlapping events**

Situations where events overlap in time, corresponding to needing more than one crane to perform a required move, may occur. This is the case if an event is to be handled before the previous one is completed, i.e. if the previous event gives rise to moving operations that exceed the interval between the events' time stamps. Three strategies concerning this issue can be adopted:

- *Accepting overlaps between events:* Reporting overlaps between events and using the information to realize if more than one crane is required to perform the operations. The advantage of this strategy is that the operations are performed as planned but there is a risk of allowing overlaps between several events, corresponding to requiring more than two cranes operating in the block which is rarely, if ever, seen in reality.
- *Accepting delays but not overlaps:* Moving the subsequent event by updating the respective time stamp to the point in time of finishing the current event. The advantage of this strategy is that situations where several cranes are needed to perform the moves are avoided whereas the disadvantage is the risk of continuously putting off events if there are small time intervals between them. However, this would indicate overtightness and infeasibility of the problem instance that could not be overcome by any manual terminal management. Furthermore, postponing one or a few events is normal practice in port container terminals and since the key time slots may be clustered, larger time intervals later in time allow catching up if delays have occurred.

- *Accepting neither overlaps nor delays:* Strict feasibility, as opposed to the less tight feasibility criteria described later (see section 3.1.4), is ensured by preventing moves that will result in overlap with the subsequent event. The advantage of applying this strategy is that both delaying operations and requiring several cranes are prevented but in return, the feasibility requirements are very rigid and may cut off good solutions where time, physical capacity, and equipment are better used.

In the algorithm the second strategy is adopted but changes are easily made such that the first one can be tested.

### 3.1.3 Evaluation of candidate positions

When a container is to be positioned, a number of candidate positions are categorized, evaluated, and graded according to their quality before the selection procedure, described in the next section, takes over. In the first step, the candidate positions are categorized according to four categorization criteria concerning the departure time of the container on top of the stack, potentially to be placed underneath the actual container. In the second step, three evaluation parameters are calculated, each reflecting a significant perspective of the position quality: The distance from the current position, the absolute difference in departure times comparing to the top stacked container, and the stack height. In the third step, the categorization criteria and evaluation parameters define the outcome of a number of penalty parameters and a quality function, controlled by these, determines the order in which the positions should be checked for feasibility by the subsequent selection procedure. The number of candidate positions, objects of the categorization, evaluation, and grading, are limited by a specified range, described below.

Throughout this section, unless otherwise specified, the container to be positioned is denoted  $c$ , its current position is  $p$  (or the arrival place if  $c$  is an incoming container), a given position from the list of candidate positions  $\mathcal{O}^C$  is denoted  $p'$ , and the container placed on top of the stack at  $p'$  is denoted  $c'$ .

#### Range

Realizing that it is rarely advantageous to place a container far from the current position, the evaluation of candidate positions is carried out within a certain range such that not all block positions are necessarily considered. The investigation can be both strictly forward-looking and combined forward- and backward-looking. Applying a forward investigation forces containers to move only in the direction of the departure place and, naturally, may eliminate advantageous moves but, on the other hand, allowing backwards moves intuitively may imply moving over longer distances, possibly resulting in more congestion and reshuffling. These settings are easily changed in the algorithm, enabling testing of the different investigation strategies. Figure 3.3 illustrates the concepts of forward and combined forward and backward investigation respectively.

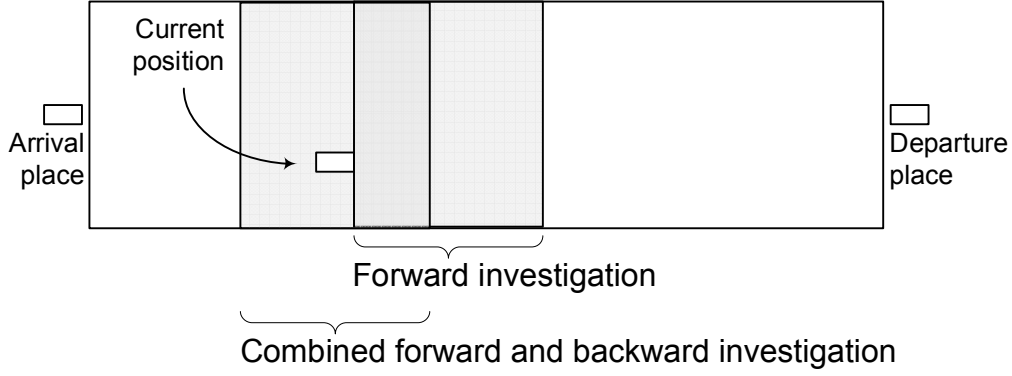


Figure 3.3: Schematic overview of the two investigation strategies: Forward-looking and combined forward- and backward-looking. In both cases, only available positions are considered.

### Categorization criteria

The first step of the evaluation procedure concerns categorization of the candidate positions within the investigation range. The criteria reflect the relationship between the departure times for container  $c$  and  $c'$  placed on top of the stack at candidate position  $p'$ . The criteria are based on the assumption that it is always best to place a container on top of one which is departing with the same transportation means, and that positioning on top of a container to depart later than the actual container is to be preferred to stacks with containers leaving before. This strategy allows containers to be ready for departure at the same time and - depending on the equipment type - to carry two containers simultaneously to their means of transportation. Furthermore, always placing container  $c$  on a stack where  $c'$  departs after  $c$  minimizes the risk of having to reshuffle  $c$  - at least on account  $c'$ . In case of an empty stack at the candidate position, it is rarely advantageous to place a container  $c$  which is relatively close to its departure time since other containers which are placed on top of it later on are likely to be departing after  $c$ . Therefore, time conflicts with containers to be placed in the future may be avoided if reserving such empty positions for containers with late departure times.

Thus, when categorizing candidate positions the departure time for container  $c$  is compared only to containers on top of the stacks. An empty stack corresponds to a dummy container  $c'$  on top with  $D_{c'}$  equal to the latest departure time, i.e. placing the container at an empty position never conflicts with lower placed containers. Clearly, all available positions within the investigation range meet one of the categorization criteria defined as:

- 0 The stack is empty, corresponding to  $D_{c'} = \max\{D_{c''} \mid c'' \in \mathcal{C}\}$ , i.e.  $D_{c'} - D_c$  is large.
- 1 Container  $c'$  on top of candidate position  $p'$  departs simultaneously with container  $c$ , i.e.  $D_{c'} - D_c = 0$ .

- 2 Container  $c'$  on top of candidate position  $p'$  departs after container  $c$ , i.e.  $D_{c'} - D_c > 0$ .
- 3 Container  $c'$  on top of candidate position  $p'$  departs before container  $c$ , i.e.  $D_{c'} - D_c < 0$ .

Clearly, criterion 1 is to be preferred to criterion 2 and positions in category 3 should always be avoided. If the equipment does not allow simultaneous moves, the second criterion may be better than the first. Whereas the relationship between criterion 1, 2, and 3 can be unambiguously stated given the properties of the system this cannot be done for criterion 0. However, as well as for criterion 1 and 2, empty stacks, falling in with category 0, are always preferred to positions belonging to category 3. The preferences are reflected in the penalty parameters, described later in connection with the quality functions.

Figure 3.4 illustrates the principle in three of the four categorization criteria, criterion 1 being the best and criterion 3 being the poorest. Criterion 0 is not represented in the figure. The time axis measures the difference in departure times for container  $c$  and  $c'$ . A positive difference,  $D_{c'} - D_c > 0$ , corresponds to  $c'$  departing after  $c$  which allows the latest placed container  $c$  to be removed to meet its departure time before it is forced by the impending departure of container  $c'$ . Thereby, unnecessary reshuffling may be prevented. A negative difference,  $D_{c'} - D_c < 0$ , corresponds to  $c'$  departing before  $c$  which, for certain, will require reshuffling of  $c$  before moving  $c'$  to the departure place.

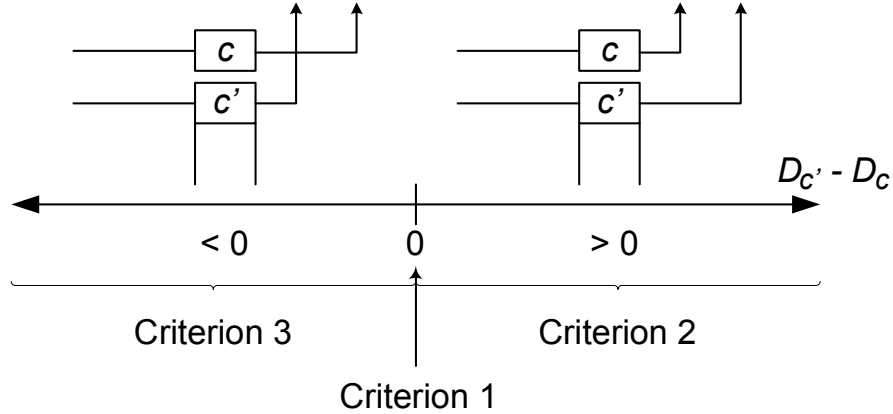


Figure 3.4: Overview of categorization criterion 1, 2, and 3 when considering placing container  $c$  at a position with container  $c'$  on top of the stack: The first criterion is that container  $c'$  departs at the same time as container  $c$ , i.e.  $D_{c'} = D_c$ , the second criterion is that container  $c'$  departs after container  $c$ , i.e.  $D_{c'} > D_c$ , and the third criterion is the opposite situation, i.e.  $D_{c'} < D_c$ . Criterion 1 is preferred to criterion 2 which again is preferred to criterion 3.

When a candidate position is assigned to one of the four categories, it is labeled  $k = 0, 1, 2$ , or  $3$  respectively. The index  $k$  is input to the quality functions described later, as are the evaluation parameters described below.

## Evaluation parameters

In the second step of the evaluation procedure three evaluation parameters which reflect the properties of the candidate position are computed. The three parameters, *dist*, *time*, and *stack*, concern the distance to the candidate position, the absolute difference in departure times of  $c$  and  $c'$ , and the stack height. These properties characterize the quality of the position - the higher value of the evaluation parameter, the poorer quality of the position. The evaluation parameters are defined as follows:

$$dist = T_{pp'},$$

$$time = |D_{c'} - D_c|, \text{ and}$$

$$stack = \text{number of containers placed at } p'.$$

The purpose of seeking out candidate positions close to the current one is to avoid unnecessary moves over long distances. The rationale for rewarding positions where the top placed container departs close to  $c$  is as follows. If evaluating two candidate positions,  $p'$  and  $p''$ , with top containers departing at  $D_{c'}$  and  $D_{c''}$  respectively, and  $D_c < D_{c'} < D_{c''}$ , by the principle of minimizing the absolute difference in departure times, position  $p'$  is preferred to  $p''$ , corresponding to reserving position  $p''$  for a subsequent container  $c^{future}$  departing later than  $c$ , e.g.  $D_{c'} < D_{c^{future}} < D_{c''}$ , thus possibly preventing future reshuffling if the alternative was placing  $c^{future}$  at  $p'$ . Also, if  $D_{c''} < D_{c'} < D_c$ , position  $p'$  is preferred since  $c$  may be close to its departure time when having to reshuffle it on account of  $c'$  and, therefore, may be moved directly to the departure place. The reason for seeking to avoid positions with high stacks is to obtain a block which is as equally leveled as possible, facilitating crane movement over the area. Clearly, this object may sometimes conflict with categorization criterion 0, reserving empty positions for certain containers with late departure times, and therefore the penalty parameters in the quality functions described below must be chosen carefully in order to obtain the desired evaluation and grading of the candidate positions.

## Quality functions

The third step of the evaluation procedure concerns grading of the candidate positions by a quality function  $f_k$ , equal to the product of three subfunctions, one per evaluation parameter *dist*, *time*, and *stack*. The subfunction  $f_k^{time}$  takes the categorization criterion  $k$  as input, enabling differentiation between positions with positive and negative departure time differences, whereas  $f^{dist}$  and  $f^{stack}$  are independent of  $k$ .

The three subfunctions are controlled by the penalty parameters,  $\alpha^{dist}$ ,  $\alpha_k^{time}$ , and  $\alpha^{stack}$ , assuming low values when the respective properties are desired and large values when they are poor. The subfunctions assume the form  $f^{dist} = e^{-\alpha^{dist} \cdot dist}$ ,  $f_k^{time} = e^{-\alpha_k^{time} \cdot time}$ , and  $f^{stack} = e^{-\alpha^{stack} \cdot stack}$ . Thus, the quality  $q_{p'}$  of candidate positions  $p'$ , belonging to category  $k$  and with computed evaluation parameters, *dist*, *time*, and *stack*, is given by the function:



$$\begin{aligned}
q_{p'} &= f^{dist} \cdot f_k^{time} \cdot f^{stack} \\
&= e^{-\alpha^{dist} \cdot dist} \cdot e^{-\alpha_k^{time} \cdot time} \cdot e^{-\alpha^{stack} \cdot stack}
\end{aligned}$$

The exponential subfunctions determine the quality related to each evaluation parameter, function values larger than a specified number  $\eta^{dist}$ ,  $\eta_k^{time}$ , and  $\eta^{stack}$  respectively, and smaller than the best possible quality of 1.0. Consequently, the quality function  $f_k$  results in quality values between  $\eta^{dist} \cdot \eta_k^{time} \cdot \eta^{stack}$  and 1.0. The minimal function values,  $\eta^{dist}$ ,  $\eta_k^{time}$ , and  $\eta^{stack}$ , are all smaller than 1.0.

The default values of the penalty parameters are determined as follows. Knowing that the lowest possible value of each evaluation parameter is 0, resulting in a quality of 1.0 regardless of the penalty parameters, these are determined by the largest values, the worst case outcome,  $dist^W = R + B - 1$ ,  $time^W = \max\{D_c \mid c \in \mathcal{C}\} - \min\{A_c \mid c \in \mathcal{C}\} = L$ , and  $stack^W = H$  respectively:

$$\begin{aligned}
\eta^{dist} &= e^{-\alpha^{dist}(R+B-1)} \quad \Rightarrow \quad \alpha^{dist} = \frac{-\ln(\eta^{dist})}{R+B-1} \\
\eta_k^{time} &= e^{-\alpha_k^{time} L} \quad \Rightarrow \quad \alpha_k^{time} = \frac{-\ln(\eta_k^{time})}{L}, \quad k = 0, 1, 2, 3 \\
\eta^{stack} &= e^{-\alpha^{stack} H} \quad \Rightarrow \quad \alpha^{stack} = \frac{-\ln(\eta^{stack})}{H}
\end{aligned}$$

Forward,  $\underline{\alpha}$  is short for  $(\alpha^{dist}, \alpha_k^{time}, \alpha^{stack})$ . When the quality stamp  $q_{p'}$  for all candidate positions  $p'$  are computed by the quality functions, they are sorted by descending value and returned to the next routine: Selection of the best possible position.

### 3.1.4 Selection of the best possible position

When all candidate positions have been categorized, evaluated, and graded, the selection routines come into play. The procedure is to check the list of qualities  $\mathcal{Q}^C$ , corresponding to the candidate positions  $p' \in \mathcal{O}^C$ , from the top and select the first feasible position. Two checks observe feasibility of the potential new position regarding container  $c$  to be placed as well as all stack containers  $c'$ , in both cases involving the terms position worst case moving time  $T_p^W$  and container worst case moving time  $T_c^W$ . The position worst case moving time is the longest transportation time to another position in the block, i.e.  $T_p^W = \max\{T_{pp'} \mid p' \in \mathcal{P}\}$ , and the container worst case moving time is the time required to reshuffle all containers placed on top of it to another position - possibly the farthest in the block - and move it to the departure place, i.e.  $T_c^W = n \cdot T_{p_c}^W + T_{p_c I}$ , where  $n$  is the number of containers placed on top of container  $c$ . Clearly, if  $c$  is the top container in the stack, its worst case moving time equals the transportation time to the departure place  $T_{p_c I}$ .

The first feasibility check concerns container  $c$ . Being at the current position  $p_c$ , it must be ensured that moving  $c$  to candidate position  $p'$  does not imply a future conflict with the departure time  $D_c$ , i.e. adding the transportation time between  $p_c$  and  $p'$  and the potential container worst case moving time  $T_c^W$ , equal to the time required to move

it to the departure place, to the current system time  $T^S$  must not exceed the departure time. If the first feasibility criterion  $c_1^F$  below is not met, position  $p'$  can be rejected on account of container  $c$ .

$$c_1^F : T^S + T_{p_c p'} + T_{p' I} \leq D_c$$

The second feasibility check concerns all stack containers  $c'$  at the candidate position. A similar observation is performed, checking if placing container  $c$  at  $p'$  will imply future conflicts with the stack containers' departure times  $D_{c'}$ . For all stack containers it must hold that updating the system time  $T^S$  by the move from  $p_c$  to  $p'$  and adding the container worst case moving time  $T_{c'}^W$  does not exceed its departure time. If the second feasibility criterion  $c_2^F$  below is not met for one of the stack containers  $c'$ , position  $p'$  is rejected on account of  $c'$ .

$$c_2^F : T^S + T_{p_c p'} + T_{c'}^W \leq D_{c'}$$

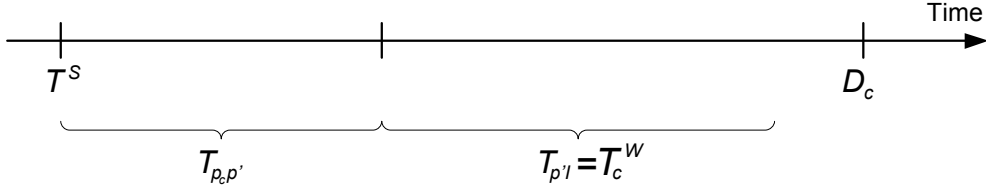
Clearly, the feasibility requirements only concern future conflicts with containers' departure times, not with future events. Furthermore, the feasibility criteria do not consider empty moves which may lead to delays in the system. This is, however, easily adjusted in case of too many delays.

If a candidate position is rejected due to failure to satisfy one or several feasibility checks the next in the list  $\mathcal{Q}^C$  is considered. In this way, the first position to meet the feasibility criteria - and thereby the best - is selected as new position for container  $c$ . The best feasible position is denoted  $p^*$ . Figure 3.5 provides an overview of the two feasibility criteria, illustrating a situation where candidate position  $p'$  is feasible. The position would have been infeasible if  $D_c$  had been earlier, lying within the right-hand side brace, and/or if this had been the case with one or several stack containers' departure time  $D_{c'}$ .

Several strategies for the position worst case moving time  $T_p^W$  can be adopted. The default value is the longest transportation time to any other position, i.e. to the position at the farthest corner of the block, but if performing only forward-looking evaluation of candidate positions and only moving containers in a forward direction in the block, it makes no sense to consider the worst case moving time towards the positions in the first bay. Therefore, the position worst case moving time, clearly found by computing the transportation times to the four corner positions,  $p^{NE}$ ,  $p^{SE}$ ,  $p^{NW}$ , and  $p^{SW}$ , can be adjusted by scaling the forward and the backward distances and, thereby, disregarding the positions in the first bay if applying a forward-looking strategy.

Furthermore, the scaling opportunities enable adjustment of the rigidity of the feasibility requirements. Defining the strict position worst case moving time as  $T_p^W = \max\{T_{pp'} \mid p' \in \mathcal{P}\} = \max\{T_{pp^{NE}}, T_{pp^{SE}}, T_{pp^{NW}}, T_{pp^{SW}}\}$  implies quite rigid feasibility criteria as, in practice, it is unlikely that all containers to be removed in order to access a lower one are repositioned to the farthest position in the block. This may lead to exclusion of high-quality positions whereas scaling the four corner transportation times may allow positionings where some stack containers to be moved to the departure place

### Feasibility check 1: container $c$ to be positioned



### Feasibility check 2: all stack containers $c'$

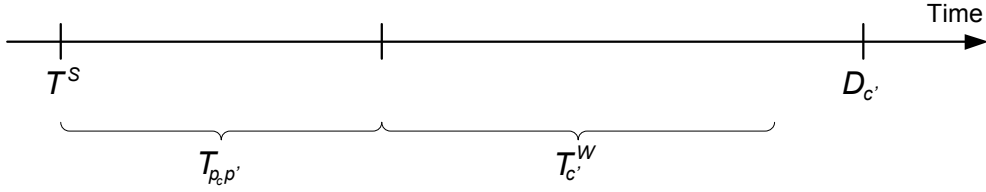


Figure 3.5: Overview of the two feasibility checks: The first concerning container  $c$  to be placed at candidate position  $p'$ , and the second concerning all containers  $c'$  in the stack at  $p'$ . Starting from the current system time  $T^S$ , the transportation time  $T_{p_c p'}$  from container  $c$ 's current position  $p_c$  to  $p'$  and the container worst case moving time -  $T_{p'I}$  for container  $c$  and  $T_{c'}^W = n \cdot T_{p_c}^W + T_{p_c I}$  for each stack container  $c'$  - are added. If this does not exceed the departure time,  $D_c$  and  $D_{c'}$  respectively, position  $p'$  is feasible and selected from the list.

later on depend on not having to reshuffle all top placed containers to the farthets corner position.

The scaling of  $T_p^W$  is illustrated in figure 3.6. The transportation times to the forward corner positions,  $p^{NE}$  and  $p^{SE}$  at the departure end of the block are scaled with the scaling factors  $\sigma^{NE}$  and  $\sigma^{SE}$  and, correspondingly, the transportation times to the backward corner positions are scaled with the factors  $\sigma^{NW}$  and  $\sigma^{SW}$ , all varying between 0.0 and 1.0, depending on the desired weight of the corner positions. Forward,  $\underline{\sigma}$  is short for  $(\sigma^{NE}, \sigma^{SE}, \sigma^{NW}, \sigma^{SW})$ .

If applying a forward-looking strategy for evaluation of candidate positions the farthest positions at the arrival end of the block can be excluded from consideration by setting  $\sigma^{NW} = \sigma^{SW} = 0.0$ . Scaling the worst case moving times by factors smaller than 1.0 naturally may imply infeasibility and delay departure times for some containers. The extend to which delays are allowed is adjusted by the scaling factors.

## 3.2 The algorithm

Based on the introcuton to the main concepts in the previous section a technical description of the algorithm is provided in this section. The following section presents the considerable amount of parameters constituting input data to the heuristic as well as

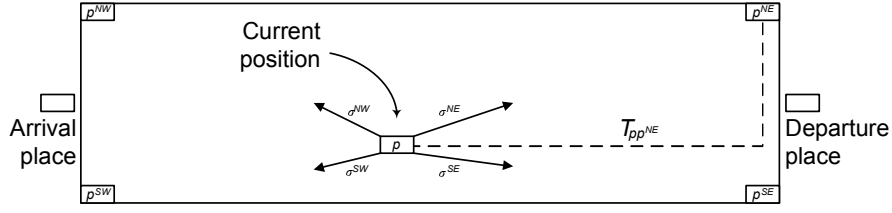


Figure 3.6: Scaling of the position worst case moving times  $T_p^W$  by the four scaling factors  $\sigma^{NW}$ ,  $\sigma^{SW}$ ,  $\sigma^{NE}$ , and  $\sigma^{SE}$ , adjusting the transportation times to the four corner positions,  $p^{NW}$ ,  $p^{SW}$ ,  $p^{NE}$ , and  $p^{SE}$ , when computing  $T_p^W$ . By setting  $\sigma^{NW} = \sigma^{SW} = 0.0$ , the farthest positions at the arrival end of the block can be excluded from consideration. Less rigid feasibility requirements can be obtained by letting the scaling factors assume values lower than 1.0.

the configuration of the problem instances is described. Subsequently, the pseudo code for the algorithm is presented and each function is explained.

### 3.2.1 Input data and configuration

The problem parameters consists of input data and parameters defined by these data. The latter can be divided into global and local parameters. Input data is displayed in table 3.1, global data in table 3.2, and local data in table 3.3.

| Input data                                           |                                                         |
|------------------------------------------------------|---------------------------------------------------------|
| <b>Problem specific parameters</b>                   |                                                         |
| $\mathcal{C} = [1, \dots, C]$                        | List of containers                                      |
| $[A_1, \dots, A_C]$                                  | List of arrival times                                   |
| $[D_1, \dots, D_C]$                                  | List of departure times                                 |
| $R$                                                  | Number of rows in the block                             |
| $B$                                                  | Number of bays in the block                             |
| $H$                                                  | Maximum stack height                                    |
| $V$                                                  | Crane velocity                                          |
| <b>Tuning parameters</b>                             |                                                         |
| $\rho$                                               | Investigation range                                     |
| $\mu$                                                | Small time interval                                     |
| $\eta^{dist}, \eta_k^{time}, \eta^{stack}$           | Minimal function values for the quality subfunctions    |
| $\alpha^{dist}, \alpha_k^{time}, \alpha^{stack}$     | Penalty parameters for the quality subfunctions         |
| $\sigma^{NE}, \sigma^{SE}, \sigma^{NW}, \sigma^{SW}$ | Scaling parameters for position worst case moving times |

Table 3.1: Notation for the input data, consisting of problem specific parameters and tuning parameters.

Part of the input data defines the actual problem instance and is referred to as constant global parameters. The remaining parameters are subject to change - the global parameters in several functions and routines and the local parameters in specific

subfunctions. The input data and function parameters are disposed in the following tables approximately in order of appearance in the algorithm.

Input data consists of a number of problem specific parameters by which the problem instance is configured and tuning parameters that can be varied in order to investigate different strategies and rules. The problem specific parameters are the lists of containers  $c$ , arrival times  $A_c$  and departure times  $D_c$ , the number of rows and bays in the block,  $R$  and  $B$ , the maximum stacking height  $H$ , and the crane velocity  $V$ . The tuning parameters are the range  $\rho$ , determining the number of candidate positions when positioning containers, a small time interval  $\mu$ , indicating if a container is close enough to its departure time to be moved to the departure place, the minimal function values and penalty parameters,  $\eta^{dist}$ ,  $\eta_k^{time}$ ,  $\eta^{stack}$ ,  $\alpha^{dist}$ ,  $\alpha_k^{time}$ , and  $\alpha^{stack}$ , in the quality functions, and the scaling parameters,  $\sigma^{NE}$ ,  $\sigma^{SE}$ ,  $\sigma^{NW}$ , and  $\sigma^{SW}$ , applied when computing the position worst case moving time. Table 3.1 provides an overview of the input data containing the problem specific parameters and the tuning parameters.

| Global data                           |                                                                                                       |
|---------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>Constant parameters</b>            |                                                                                                       |
| $\mathcal{P} = [1, \dots, R \cdot B]$ | List of positions $p$                                                                                 |
| $\{0, I\}$                            | Arrival and departure place                                                                           |
| $r_i$                                 | Row co-ordinate for place/position $i$                                                                |
| $b_i$                                 | Bay co-ordinate for place/position $i$                                                                |
| $T_{ij}$                              | Transportation time between nodes $i \neq I$ and $j \neq 0$                                           |
| $T_p^W$                               | Position worst case moving time for $p$                                                               |
| $L$                                   | Length of planning horizon: $\max\{D_c \mid c \in \mathcal{C}\} - \min\{A_c \mid c \in \mathcal{C}\}$ |
| $\mathcal{E}^A$                       | List of arrival events                                                                                |
| $\mathcal{E}^D$                       | List of departure events                                                                              |
| <b>Dynamically changed parameters</b> |                                                                                                       |
| $\mathcal{E}^{PD}$                    | List of predeparture events                                                                           |
| $\mathcal{E}$                         | List of events: $\mathcal{E} = \mathcal{E}^A \cup \mathcal{E}^{PD} \cup \mathcal{E}^D$                |
| $e_c \in \mathcal{E}$                 | Event concerning container $c$                                                                        |
| $c_e$                                 | Event container for event $e_c$                                                                       |
| $T^S$                                 | System time                                                                                           |
| $\mathcal{S}_p$                       | Current list of containers in the stack at position $p$                                               |
| $\mathcal{O}$                         | Current list of available positions in the block                                                      |
| $\mathcal{O}^C \subseteq \mathcal{O}$ | Current list of candidate positions                                                                   |
| $\mathcal{Q}^C$                       | Current list of quality values for candidate positions                                                |
| $\mathcal{D}^E$                       | List of early departed containers                                                                     |
| $\mathcal{D}^O$                       | List of on-time departed containers                                                                   |
| $\mathcal{D}^L$                       | List of late departed containers                                                                      |

Table 3.2: Notation for the global data, consisting of constant and variable parameters.

The global data are accessible in all routines in the algorithm and consist of a number of constant parameters which constitute the problem instance based on the input

data and a number of variable parameters which are changed by various procedures throughout the algorithm.

A problem instance is configured as follows. The list of positions  $\mathcal{P} = [1, \dots, R \cdot B]$  is based on the number of rows and bays, the arrival place is numbered 0, and the departure place is  $I = R \cdot B + 1$ . Each position  $p$  is assigned a set of row and bay co-ordinates  $(r_p, b_p)$ , corresponding to the row and bay number. There are  $R$  positions in each bay. For the arrival and departure place,  $(r_0, b_0) = (\lceil R/2 \rceil, 0)$  and  $(r_I, b_I) = (\lceil R/2 \rceil, B + 1)$ . The transportation time  $T_{ij}$  between two positions/places,  $i$  and  $j$ , is, equivalent to the mathematical models, the manhattan distance  $|r_{p'} - r_p| + |b_{p'} - b_p|$  divided by the crane velocity  $V$ . Note that the time for hoisting and lowering containers is included in the transportation which may distort the actual transportation times as empty and full moves over the same distance are assumed equal. Each position is assigned a worst case moving time  $T_p^W = \max\{T_{pp'} \mid p' \in \mathcal{P}\}$ , equal to the transportation time to the farthest corner position in the block. The length of the planning horizon is computed as  $L = \max\{D_c \mid c \in \mathcal{C}\} - \min\{A_c \mid c \in \mathcal{C}\}$ . Finally, the lists of arrival and departure events,  $\mathcal{E}^A$  and  $\mathcal{E}^D$ , also remaining constant throughout the algorithm, are computed.

The dynamic parameters are the list of predeparture events  $\mathcal{E}^{PD}$ , together with  $\mathcal{E}^A$  and  $\mathcal{E}^D$  constituting the list of events  $\mathcal{E}$ , the system time  $T^S$ , initially set to the earliest arrival time  $\min\{A_c \mid c \in \mathcal{C}\}$ , corresponding to the first event, the stack at each position  $\mathcal{S}_p$ , initially empty, the list of available positions in the block  $\mathcal{O}$ , initially equal to  $\mathcal{P}$ , the lists of candidate positions and quality values,  $\mathcal{O}^C$  and  $\mathcal{Q}^C$ , used when positioning containers, and the lists of departure time accuracy,  $\mathcal{D}^E$  for early,  $\mathcal{D}^O$  for on-time, and  $\mathcal{D}^L$  for late departed containers.

Table 3.2 states the global parameters, some constant throughout the algorithm and some changed by functions and subfunctions.

| Local data                            |                                                    |
|---------------------------------------|----------------------------------------------------|
| <b>Parameters in subfunction A</b>    |                                                    |
| $k = 0, 1, 2, 3$                      | Categorization criteria                            |
| $dist, time, stack$                   | Evaluation parameters                              |
| $f_k$                                 | Quality function for categorization criterion $k$  |
| $f_k^{dist}, f_k^{time}, f_k^{stack}$ | Quality subfunctions for the evaluation parameters |
| <b>Parameters in subfunction B</b>    |                                                    |
| $T_c^W$                               | Current worst case moving time for container $c$   |
| $c_1^F, c_2^F$                        | Feasibility criteria 1 and 2                       |
| $p^*$                                 | Current best position for a given container        |

Table 3.3: Notation for the local function parameters for subfunctions A and B.

The local data is only accessible to subfunctions A and B, performing the evaluation and selection procedures when positioning a container. The parameters in subfunction A are the categorization criteria,  $k = 0, 1, 2, 3$ , the evaluation parameters,  $dist$ ,  $time$ , and  $stack$ , and the quality function  $f_k$ , consisting of three quality subfunctions,  $f_k^{dist}$ ,  $f_k^{time}$ , and  $f_k^{stack}$ , corresponding to the categorization criterion  $k$  and the three evaluation

parameters. The parameters in subfunction B are the worst case moving time  $T_c^W$  for a container  $c$ , the two feasibility criteria,  $c_1^F$  and  $c_2^F$ , and the best position  $p^*$  found for the container to be positioned. Table 3.3 contains the local parameters for subfunctions A and B.

### 3.2.2 Pseudo code

The algorithm consists of two steps: An initialization step and an event processing step. In the main step a number of functions are applied, all described in detail below. If not otherwise specified, data is global. The following representation is a high-level overview of the algorithm and the argument lists illustrate the key features of the functions rather than the complete code implemented.

#### Step 0: initialization

In the first step the system time  $T^S$  is set to the smallest time stamp, i.e. the earliest arrival time  $\min\{A_c \mid c \in \mathcal{C}\}$ , the worst case moving time for each position  $T_p^W$  is set to the distance to the remotest corner position,  $\max\{T_{pp'} \mid p' \in \mathcal{P} \setminus \{p\}\}$ , the stack at each position  $\mathcal{S}_p$  is set to an empty list, the list of free positions  $\mathcal{O}$ , clearly including all positions in an empty block, is set equal to  $\mathcal{P}$ , the departure time accuracy,  $\mathcal{D}^E$  for early departures,  $\mathcal{D}^O$  for on-time departures, and  $\mathcal{D}^L$  for late departures, are set to empty lists, and the list of events  $\mathcal{E}$  is created, equal to arrival and departure times for the containers. Finally, the list of events is sorted by ascending event time.

##### Step 0: initialization

```

let  $T^S = \min\{A_c \mid c \in \mathcal{C}\}$ 
let  $T_p^W = \max\{T_{pp'} \mid p' \in \mathcal{P} \setminus \{p\}\} \forall p \in \mathcal{P}$ 
let  $\mathcal{S}_p = \emptyset$ 
let  $\mathcal{O} = \mathcal{P}$ 
let  $\mathcal{D}^E = \mathcal{D}^O = \mathcal{D}^L = \emptyset$ 
let  $\mathcal{E} = \mathcal{E}^A \cup \mathcal{E}^{PD} \cup \mathcal{E}^D = \{e_c^A \mid c \in \mathcal{C}\} \cup \{e_c^{PD} \mid c \in \mathcal{C}\} \cup \{e_c^D \mid c \in \mathcal{C}\}$ ,
    where  $e_c^A = A_c$ ,  $e_c^{PD} = D_c$ ,  $e_c^D = D_c$ ,  $\forall c \in \mathcal{C}$ 
sort  $\mathcal{E}$  by ascending event time

```

#### Step 1: process events

Step 1 constitutes the main loop of the algorithm. For all events in the list  $\mathcal{E}$ , first, a time check is performed, adjusting either the event time or the system time, and secondly, one of the functions `positionIncomingContainer(c)`, `repositionToBufferZone(c)`, or `removeOutgoingContainer(c)`, is called, depending on the type of event.

The time check ensures that the event time  $e_c$  is put forward to the system time  $T^S$  if delays have caused  $T^S$  to exceed the planned time for the event's operations, i.e. if the previous event finished too late for the actual one to start on time. On the other hand, if no delays have occurred, i.e. if the event is scheduled to take place

immediately or in the future, the system time is not put forward at this stage. This allows the crane to perform any empty moves and/or repositionings that may be required to access the event container in the time between finishing the previous event and starting handling the container of the current event. This procedure corresponds to a single crane system where operations cannot be performed simultaneously. If instead two cranes are operating at the block, moves can be performed in parallel, which could be represented by two separate system times,  $T_1^S$  and  $T_2^S$ , allowing one of the cranes to skip an event if it finished the latest operations too late and the other crane is available. This would, however, increase complexity since it would have to be decided which crane should perform each operation. Alternatively, if neither delays nor overlaps are accepted, time conflicts could result in rejection of jobs which, in practice, corresponds to not storing an incoming container or not moving a container towards the departure place. In this algorithm, we assume one crane available and apply the first strategy: Accepting time overlaps, potentially necessitating delays of some events.

The subsequent processing of the events calls function `positionIncomingContainer( $c$ )` in case of an arrival event, function `repositionToBufferZone( $c$ )` in case of predeparture event, and function `removeOutgoingContainer( $c$ )` in case of a departure event. Finally, after removing the processed event  $e_c$ , the list of events  $\mathcal{E}$  is sorted.

**Step 1: process events**

```

for all  $e_c \in \mathcal{E}$  do
  if  $T^S > e_c$  then
     $e_c = T^S$ 
  if  $e_c \in \mathcal{E}^A$  then
    positionIncomingContainer( $c$ )
  if  $e_c \in \mathcal{E}^{PD}$  then
    repositionToBufferZone( $c$ )
  if  $e_c \in \mathcal{E}^D$  then
    removeOutgoingContainer( $c$ )
 $\mathcal{E} = \mathcal{E} \setminus \{e_c\}$ 
sort  $\mathcal{E}$  by ascending event time

```

**Function 1: positionIncomingContainer( $c$ )**

Function `positionIncomingContainer( $c$ )` handles containers to be removed from the arrival place and positioned in the block. The list of candidate positions  $\mathcal{O}^C$  is initially set to the first  $\rho$  members of the list of available positions  $\mathcal{O}$ , where  $\rho$  is the specified search range. Then the two central functions in the algorithm take effect: Function `evaluateCandidatePositions( $\mathcal{O}^C, c$ )` returns a prioritized list of quality values  $\mathcal{Q}^C$  and `selectMostFavourablePosition( $\mathcal{Q}^C, c$ )` returns the best position  $p^*$  from the candidate list. If such a position is found by the above routines the container move is handled by functions `removeFromCurrentPosition( $c$ )` and `moveToNewPosition( $p^*, c$ )`, and if no



feasible position is found within the search range the container is moved to the buffer zone at the farthest end of the block by function `repositionToBufferZone(c)`.

**Function 1:** `positionIncomingContainer(c)`

```

 $\mathcal{O}^C = \{p \in \mathcal{O} \mid p < \rho\}$ 
 $\mathcal{Q}^C = \text{evaluateCandidatePositions}(\mathcal{O}^C, c)$ 
 $p^* = \text{selectMostFavourablePosition}(\mathcal{Q}^C, c)$ 
if  $p^* \neq \emptyset$  then
    removeFromCurrentPosition(c)
    moveToNewPosition(p*, c)
else
    repositionToBufferZone(c)

```

**Function 2:** `repositionToBufferZone(c)`

Function `repositionToBufferZone(c)` is called in two situations: In case of a predeparture event, concerning containers about to depart from the terminal, and in the case that no feasible position is found within the search range when positioning an incoming container or when reshuffling.

When moving container  $c$  from the stack at position  $p_c$ , one or more containers may be placed on top of it. If so, these are removed by function `repositionTopStackContainers(c)` before handling the actual one. We recall that reshuffling and/or empty moves, required to access the event container, may be performed in the time interval between finishing the previous event and starting the actual one, if the events are not immediately successive.

When container  $c$  is the topmost container at position  $p_c$ , the system time  $T^s$ , if not already equal to  $e_c$ , is put forward to the event time. Then, it is checked if moving  $c$  directly to the departure place will result in the container being there either later, right on time, or slightly earlier than the scheduled departure time  $D_c$ , in which case it is clearly not reasonable to reshuffle it to the buffer zone first. Instead, the container is immediately removed from the block by function `removeOutgoingContainer(c)`, also registering whether it is late, on time, or early (see page 82). Thus, if performing the move will exceed the container's departure time, i.e. if  $T^s + T_{p_c I} > D_c$ , it is moved to the departure place by function `removeOutgoingContainer(c)` and added to the list of late departures  $\mathcal{D}^L$ . Correspondingly, if moving the container to the departure place meets the departure time, i.e. if  $T^s + T_{p_c I} = D_c$ , it is also removed and now added to the list of on-time departures  $\mathcal{D}^O$ . Finally, if the move will result in an early but close to on-time departure, i.e. if  $T^s + T_{p_c I} > D_c - \mu$ , container  $c$  is also removed and the list of early departures  $\mathcal{D}^E$  is updated to count  $c$  as well.

If none of the above situations occur the procedures correspond to the ones described in function `positionIncomingContainer(c)`. Initially, the list of candidate positions  $\mathcal{O}^C$  is set to the last instead of the first  $\rho$  members of  $\mathcal{O}$ . Subsequently, the two functions, `evaluateCandidatePositions( $\mathcal{O}^C, c$ )` and `selectMostFavourablePosition( $\mathcal{Q}^C, c$ )`, returning

$\mathcal{Q}^C$  and  $p^*$  respectively, are called. If a feasible position  $p^*$  is found, the move from the actual position  $p_c$  to  $p^*$  is handled by function  $\text{moveToNewPosition}(p^*, c)$  after registering the removal from  $p_c$  by function  $\text{removeFromCurrentPosition}(c)$ . If, on the other hand, no feasible position is found in the buffer zone, the container is moved directly to the departure place by function  $\text{removeOutgoingContainer}(c)$  in which the early departure is registered in the list  $\mathcal{D}^E$ .

**Function 2:**  $\text{repositionToBufferZone}(c)$

```

repositionTopStackContainers( $c$ )
if  $T^S < e_c$  then
     $T^S = e_c$ 
if  $T^S + T_{p_c I} > D_c$  then
    removeOutgoingContainer( $c$ )
if  $T^S + T_{p_c I} = D_c$  then
    removeOutgoingContainer( $c$ )
if  $T^S + T_{p_c I} > D_c - \mu$  then
    removeOutgoingContainer( $c$ )
if  $T^S + T_{p_c I} < D_c - \mu$  then
     $\mathcal{O}^C = \{p \in \mathcal{O} \mid p > \text{last}(\mathcal{O}) - \rho\}$ 
     $\mathcal{Q}^C = \text{evaluateCandidatePositions}(\mathcal{O}^C, c)$ 
     $p^* = \text{selectMostFavourablePosition}(\mathcal{Q}^C, c)$ 
    if  $p^* \neq \emptyset$  then
        removeFromCurrentPosition( $c$ )
        moveToNewPosition( $p^*, c$ )
    else
        removeOutgoingContainer( $c$ )

```

**Function 3:**  $\text{removeOutgoingContainer}(c)$

Function  $\text{removeOutgoingContainer}(c)$  is called in three types of situations: In case of a departure event, when lack of feasible positions enforces a move to the departure place, and in case that a move to the buffer zone shows disadvantageous due to approaching or exceeded departure time.

Since earlier events and operations may have resulted in moving container  $c$  to the departure place before its departure event occurs, a first check observes if the container has already been removed from the system, i.e. if it is contained in one of the three lists,  $\mathcal{D}^E$ ,  $\mathcal{D}^O$ , or  $\mathcal{D}^L$ . If this is not the case, containers placed on top of it in the stack, if any, are repositioned by function  $\text{repositionTopStackContainers}(c)$  before the system time is set to the event time and container  $c$  is removed from position  $p_c$  by function  $\text{removeFromCurrentPosition}(c)$ . Note that if container  $c$  is positioned at the buffer zone after its predeparture event in order to be ready for the approaching departure time it is unlikely that there are containers placed on top of it, unless departing immediately

before  $c$ , due to the short time available to remove them again before handling container  $c$ . In practice, no containers will be placed on top of a container which is soon to depart.

Finally, the system time  $T^S$  is updated by the transportation times, first for the empty move from the previous end position  $p^{prev}$  and next for the full move from the current position  $p_c$  to the departure place  $I$ , and container  $c$  is inserted into  $\mathcal{D}^E$ ,  $\mathcal{D}^O$ , or  $\mathcal{D}^L$ , depending on the actual departure time.

**Function 3:** removeOutgoingContainer( $c$ )

```

if  $c \notin \mathcal{D}^E \cup \mathcal{D}^O \cup \mathcal{D}^L$  then
    repositionTopStackContainers( $c$ )
    if  $T^S < e_c$  then
         $T^S = e_c$ 
    removeFromCurrentPosition( $c$ )
     $T^S = T^S + T_{p^{prev}p}$ 
     $T^S = T^S + T_{p_c I}$ 
    if  $T^S < D_c$  then
         $\mathcal{D}^E = \mathcal{D}^E \cup \{c\}$ 
    if  $T^S = D_c$  then
         $\mathcal{D}^O = \mathcal{D}^O \cup \{c\}$ 
    if  $T^S > D_c$  then
         $\mathcal{D}^L = \mathcal{D}^L \cup \{c\}$ 

```

**Function A:** evaluateCandidatePositions( $\mathcal{O}^C, c$ )

Function evaluateCandidatePositions( $\mathcal{O}^C, c$ ) comes into play when a container  $c$  is to be positioned. The function is called by positionIncomingContainer( $c$ ) when a container is first positioned after its arrival, by repositionToBufferZone( $c$ ) when a container is reshuffled to the departure end of the block, and by repositionCloseToCurrentPosition( $c'$ ) when containers are removed from a stack in order to access a lower placed one. The only difference between these three situations is the candidate positions, limited by the search range  $\rho$ , to evaluate and choose between. When positioning an incoming container, the first  $\rho$  available positions are considered, when reshuffling to the buffer zone, the last  $\rho$  available positions are investigated, and when repositioning close to the current position, the choice is, obviously, between the closest  $\rho$  available positions.

The function contains three steps: Categorization according to the four criteria,  $k = 0, 1, 2, 3$ , calculation of values for the three evaluation parameters,  $dist$ ,  $time$ , and  $stack$ , and grading by the quality  $q_{p'}$  which is computed by the quality functions  $f_k^{dist}$ ,  $f_k^{time}$ , and  $f_k^{stack}$ , controlled by the penalty parameters  $\alpha^{dist}$ ,  $\alpha_k^{time}$ , and  $\alpha^{stack}$ . The categorization criteria concern the difference in departure times  $D_{c'} - D_c$ , reflecting if container  $c$  departs before, after, or simultaneously with  $c'$  on top of the stack, or if the stack is empty. The three evaluation parameters represent the distance to the candidate position  $p'$ , the absolute difference between departure times for the top placed container  $c'$  and  $c$ , and the stack height at  $p'$ . The categorization criteria and evaluation

**Function A:** evaluateCandidatePositions( $\mathcal{O}^C, c$ )

```

for all  $p' \in \mathcal{O}^C$  do
   $c' = \text{top}(\mathcal{S}_{p'})$ 
   $q_{p'} = 0$ 
  if  $\mathcal{S}_{p'} = \emptyset$  then
     $k = 0$ 
  if  $D_{c'} - D_c = 0$  then
     $k = 1$ 
  if  $D_{c'} - D_c > 0$  then
     $k = 2$ 
  if  $D_{c'} - D_c < 0$  then
     $k = 3$ 
   $dist = T_{p_c p'}$ 
  if  $\mathcal{S}_{p'} \neq \emptyset$  then
     $time = |D_{c'} - D_c|$ 
  else
     $time = |\max\{D_{c'} \mid c' \in \mathcal{C}\} - D_c|$ 
   $stack = \text{size}(\mathcal{S}_{p'})$ 
   $q_{p'} = e^{-\alpha^{dist}} \cdot e^{-\alpha_k^{time}} \cdot e^{-\alpha^{stack}}$ 
   $\mathcal{Q}^C = \mathcal{Q}^C \cup q_{p'}$ 
sort  $\mathcal{Q}^C$  by descending quality  $q_{p'}$ 

```

parameters, together with the penalty parameters determine the quality of each position, enabling a grading of the candidates.

The evaluation process is carried out for all positions  $p'$  in the list of candidate positions  $\mathcal{O}^C$ . Initially, the quality  $q_{p'}$  of  $p'$  is set to 0. Then, the position is checked for matching one of the categorization criteria, 0, 1, 2, and 3: Criterion 0 in case of an empty stack, criterion 1 in case of identical departure times  $D_{c'}$  and  $D_c$ , criterion 2 in case of container  $c$  departing before  $c'$ , and criterion 3 in case of container  $c$  departing after  $c'$ . Depending on which category  $p'$  belongs in, it is assigned a stamp  $k = 0, 1, 2$ , or 3. Next,  $dist$  is set to the distance, corresponding to the transportation time  $T_{p_c p'}$  to  $p'$ ,  $time$  is set to the absolute difference in departure times  $|D_{c'} - D_c|$  if container  $c'$  is placed on top of the stack and  $|\max\{D_{c'} \mid c' \in \mathcal{C}\} - D_c|$  if the stack is empty, and  $stack$  is set to the size of  $\mathcal{S}_{p'}$ . After determination of  $k$ ,  $dist$ ,  $time$ , and  $stack$  for position  $p'$ , its quality  $q_{p'}$  is computed by the quality functions  $f^{dist} = e^{-\alpha^{dist}}$ ,  $f_k^{time} = e^{-\alpha_k^{time}}$ , and  $f^{stack} = e^{-\alpha^{stack}}$ , resulting in large values if the penalty parameters,  $\alpha^{dist}$ ,  $\alpha_k^{time}$ , and  $\alpha^{stack}$ , and the evaluation parameters,  $dist$ ,  $time$ , and  $stack$ , are low and poor values if they are large. Finally,  $q_{p'}$  is added to the list of quality values  $\mathcal{Q}^C$  and after processing each  $p'$  in  $\mathcal{O}^C$ ,  $\mathcal{Q}^C$  is sorted by descending quality.

**Function B: selectMostFavourablePosition( $\mathcal{Q}^C, c$ )**

Function selectMostFavourablePosition( $\mathcal{Q}^C, c$ ), returning the best possible position  $p^*$ , handles the feasibility check of candidate positions before moving a container. Therefore, it always follows function evaluateCandidatePositions( $\mathcal{O}^C, c$ ), returning the quality values  $q_{p'} \in \mathcal{Q}^C$  for each candidate position  $p' \in \mathcal{O}^C$ . The function consists of two feasibility checks: The first concerning container  $c$  to be positioned, the second concerning all containers  $c'$  in the stack at the candidate position.

Initially no position  $p^*$  exists. In order to get the best possible position the list  $\mathcal{Q}^C$  is investigated from the top and the function stops as soon as the first feasible position is found. As long as  $p^*$  does not exist, all positions  $p'$  associated with the quality values  $q_{p'}$ , undergo the two feasibility checks. First, it is checked if the system time  $T^S$  plus the potential transportation time  $T_{p_c p'}$  plus the required time to move  $c$  from  $p'$  to the departure place  $T_{p'I}$  exceeds the departure time  $D_c$ . If this is not the case, i.e. placing  $c$  at position  $p'$  does not imply a conflict with its departure time, the second feasibility check, concerning each stack container  $c'$ , is carried out. The worst case moving time  $T_{p'}^W$  for  $c'$  is calculated as  $n \cdot T_{p'}^W + T_{p'I}$  where  $n$  is the number of containers placed on top of  $c'$ , all associated with the position worst case moving time  $T_{p'}^W$ , and the last term corresponds to the time required to move the potentially new top container  $c$  to the departure place  $T_{p'I}$ . Based on this it is checked if the system time  $T^S$  plus the time  $T_{p_c p'}$  it takes to move  $c$  to  $p'$  plus the worst case moving time  $T_{p'}^W$  exceed the departure time  $D_{c'}$ . If this is not the case, i.e. if none of the stack containers are hindered from meeting their departure time if placing  $c$  in the stack, candidate position  $p'$  is selected as the best possible and the function returns it as  $p^*$ .

**Function B: selectMostFavourablePosition( $\mathcal{Q}^C, c$ )**

```

 $p^* = \emptyset$ 
for all  $q_{p'} \in \mathcal{Q}^C$  do
  while  $p^* = \emptyset$  do
    if  $T^S + T_{p_c p'} + T_{p'I} \leq D_c$  then
      for all  $c' \in \mathcal{S}_{p'}$  do
         $T_{c'}^W = n \cdot T_{p'}^W + T_{p'I}$ 
      if  $T^S + T_{p_c p'} + T_{c'}^W \leq D_{c'}$  for all  $c' \in \mathcal{S}_{p'}$  then
         $p^* = p'$ 

```

**Function C: removeFromCurrentPosition( $c$ )**

Function removeFromCurrentPosition( $c$ ) succeeds the functions evaluateCandidatePositions( $\mathcal{O}^C, c$ ) and selectMostFavourablePosition( $\mathcal{Q}^C, c$ ) and precedes function moveToNewPosition( $p^*, c$ ) whenever a container is to be positioned. First, container  $c$  is removed from the stack  $\mathcal{S}_{p_c}$  and if the stack was full before removing  $c$ ,  $p_c$  is added to the list of available positions  $\mathcal{O}$ . Then, the worst case moving time  $T_{c_{p_c}}^W$  for each container in the stack is updated to  $(n - 1) \cdot T_{p'}^W + T_{p'I}$ , where  $n$  is the number of containers placed on

top of it, and the predeparture events are re-calculated by subtracting the worst case moving times from the respective departure times.

**Function C:** removeFromCurrentPosition( $c$ )

```

 $\mathcal{S}_{p_c} = \mathcal{S}_{p_c} \setminus \{c\}$ 
if size( $\mathcal{S}_{p_c}$ ) =  $H - 1$  then
     $\mathcal{O} = \mathcal{O} \cup p_c$ 
for all  $c_{p_c} \in \mathcal{S}_{p_c}$  do
     $T_{c_{p_c}}^W = (n - 1) \cdot T_{p'}^W + T_{p'I}$ 
     $e_{c_{p_c}}^{PD} = D_{c_{p_c}} - T_{c_{p_c}}^W$ 

```

**Function D:** moveToNewPosition( $p^*, c$ )

Function moveToNewPosition( $p^*, c$ ), always following functions evaluateCandidatePositions( $\mathcal{O}^C, c$ ), selectMostFavourablePosition( $\mathcal{Q}^C, c$ ), and removeFromCurrentPosition( $c$ ), handles the move of container  $c$  from its current position  $p_c$  to the new position  $p^*$ . First, the stack  $\mathcal{S}_{p^*}$  at position  $p^*$  and container  $c$ 's worst case moving time  $T_c^W$  are updated,  $\mathcal{S}_{p^*}$  to include  $c$  and  $T_c^W$  to equal the time required to move to the departure place. If placing  $c$  at  $p^*$  results in a full stack,  $p^*$  is removed from the list of available positions  $\mathcal{O}$ . Subsequently, the predeparture events for all containers  $c_{p^*}$  in the stack are updated by subtracting their worst case moving time, updated when selecting  $P^*$  in function selectMostFavourablePosition( $\mathcal{Q}^C, c$ ), from their departure time. Finally, the system time  $T^S$  is put forward by the transportation time from from the previous position  $p^{prev}$  to the current  $p_c$  and from  $p_c$  to the new position  $p^*$ .

**Function D:** moveToNewPosition( $p^*, c$ )

```

 $\mathcal{S}_{p^*} = \mathcal{S}_{p^*} \cup \{c\}$ 
 $T_c^W = T_{p^*I}$ 
if size( $\mathcal{S}_{p^*}$ ) =  $H$  then
     $\mathcal{O} = \mathcal{O} \setminus p^*$ 
for all  $c_{p^*} \in \mathcal{S}_{p^*}$  do
     $e_{c_{p^*}}^{PD} = D_{c_{p^*}} - T_{c_{p^*}}^W$ 
 $T^S = T^S + T_{p^{prev}p_c}$ 
 $T^S = T^S + T_{p_cp^*}$ 

```

**Function E:** repositionTopStackContainers( $c$ )

Function repositionTopStackContainers( $c$ ) handles containers, denoted  $c'$ , stacked on top of one, denoted  $c$ , to be moved from a specific position  $p_c$ . The function is only called in connection with an event concerning container  $c$ . Repeatedly, the top placed containers are removed from  $p_c$  by function repositionCloseToCurrentPosition( $c'$ ) until container  $c$  is on top and ready to be handled.

**Function E:** repositionTopStackContainers( $c$ )

```

 $c' = \text{top}(\mathcal{S}_{p_c})$ 
while  $c' \neq c$  do
    repositionCloseToCurrentPosition( $c'$ )
 $c' = \text{top}(\mathcal{S}_p)$ 

```

**Function F:** repositionCloseToCurrentPosition( $c'$ )

Function repositionCloseToCurrentPosition( $c'$ ) applies to containers  $c'$  to be repositioned from position  $p_c$  on account of a lower placed container  $c$  close to its departure time. The function is only called by function repositionTopStackContainers( $c$ ) where it is applied iteratively on all containers placed on top of  $c$ . The procedures resemble the ones used when positioning an incoming container or repositioning a container to the buffer zone. First, the list of candidate positions  $\mathcal{O}^C$  is set to a number of positions close to  $p_c$ , counting  $\rho/2$  backwards and  $\rho/2$  forwards in the list of available positions  $\mathcal{O}$ , making sure that  $\mathcal{O}^C$  does not exceed  $\mathcal{P}$ . Then, the prioritized list of quality values  $\mathcal{Q}^C$  is generated by function evaluateCandidatePositions( $\mathcal{O}^C, c'$ ) and the best possible position  $p^*$  is identified by function selectMostFavourablePosition( $\mathcal{Q}^C, c$ ). Finally, if a feasible  $p^*$  is found, container  $c'$  is removed from  $p_c$  by function removeFromCurrentPosition( $c'$ ) and moved to  $p^*$  by function moveToNewPosition( $p^*, c'$ ). If no feasible  $p^*$  is found, repositionToBufferZone( $c'$ ) attempts to move container  $c'$  to a position near the departure place.

**Function F:** repositionCloseToCurrentPosition( $c'$ )

```

 $\mathcal{O}^C = \{p \in \mathcal{O} \mid p > \max\{1, p_{c'} - \rho/2\}, p < \min\{R \cdot B, p_{c'} + \rho/2\}\}$ 
 $\mathcal{Q}^C = \text{evaluateCandidatePositions}(\mathcal{O}^C, c')$ 
 $p^* = \text{selectMostFavourablePosition}(\mathcal{Q}^C, c')$ 
if  $p^* \neq \emptyset$  then
    removeFromCurrentPosition( $c'$ )
    moveToNewPosition( $p^*, c'$ )
if  $p^* = \emptyset$  then
    repositionToBufferZone( $c'$ )

```

### 3.3 Improvements

This section discusses some perspectives for improving the basic heuristic algorithm, presented in this chapter. An often applied approach when solving problems by construction heuristics is, first, producing an initial solution by the greedy approach and, next, improving the solution by one or several improvement routines.

The basic heuristic algorithm leaves some issues unattended which may be dealt with by changing or adding procedures in the existing implementation or by constructing an

entirely new set of routines, building on the initial solution, found by the basic greedy algorithm.

In general, three critical issues are important to address, some of them causing “theoretical infeasibility”, i.e. not possible from a modeling or exact point of view, and some of them causing “practical infeasibility”, i.e. not acceptable in real-life situations:

- 1 In the present heuristic algorithm, containers can be rejected, i.e. not placed in the block at all, if no feasible positions are found. This is a theoretical as well as practical infeasibility, as all containers, entering the system, naturally must be positioned.
- 2 The heuristic in its present form allows containers to be early or late for departure, the first problem, however, overcome by setting  $\mu = 0$ . This corresponds to a theoretical infeasibility as well and, in practice, it may cause undesirable solutions.
- 3 Given the existing heuristic procedures, some unnecessary reshuffling may occur, e.g. removing a container from a position immediately after placing it there in case of a lower placed container’s predeparture time. This issue creates neither theoretical nor practical infeasibility but it is an important issue from an optimization point of view and depending on the port authorities it may be critical to avoid.

### Avoiding rejections

In order to attack the problem of possibly rejecting containers, the procedure for choosing a position for a container  $c$ , described in function `selectMostFavourablePosition( $\mathcal{Q}^C, c$ )` on page 84, now with the option of not selecting one of the candidate positions  $p'$ , must be changed. Clearly, the first feasibility criterion must be fulfilled in order to place container  $c$  at position  $p'$ , moving  $c$  directly to the departure place otherwise being the only reasonable operation. On the other hand, if the second feasibility criterion concerning the stack containers at  $p'$  is not fulfilled for any of the candidate positions, it may be reasonable - or even necessary - to place  $c$  at one of the positions anyway, partly since the time conflict may be insignificantly small and partly since the feasibility criterion is very strict as it builds on the worst case moving times rather than actual moving times.

Instead of rejecting container  $c$  based on infeasibility caused by one or several stack containers at each candidate position, the procedure is changed so that the choice falls on the position, inflicting the smallest time conflict in terms of delaying one or several stack containers. Thus, when performing the second feasibility check the maximum time conflict for each position - corresponding to the stack container which is possibly being delayed the most if placing  $c$  at  $p'$  - is registered so that the position with the smallest maximum time conflict may be selected.

This approach is implemented as described in section 3.3.1, leaving the alternative strategy of not allowing rejections an option in the basic heuristic algorithm.



## Reducing delays

The issue of containers being delayed for their scheduled departure can be addressed by analyzing a solution, provided by the basic heuristic algorithm, and searching for ways to improve it by changing certain operations.

The concept of processing events one at a time implies some potential disadvantages as completing an event without regard for subsequent ones, containing moves that may be performed “during the event”, possibly leads to delays of subsequent handled containers. As an example, consider the two events  $e_i$  and  $e_j$ , concerning the departure of container  $c_i$  and  $c_j$  with departure times  $D_i$  and  $D_j$  respectively. One container is placed on top of  $c_i$  and  $c_j$  is placed on top of its stack, implying that  $e_i$  is scheduled before  $e_j$  due to the worst case moving time for  $c_i$  even though  $D_i > D_j$ . Assume that the event preceding  $e_i$  is completed well in advance of the scheduled start time for  $e_i$ , allowing the container on top of  $c_i$  to be removed long before needing to perform the outgoing move of  $c_i$ . As  $e_i$  is completed before processing  $e_j$ , the crane idle time between the required reshuffle and the scheduled outgoing move, corresponding to the time interval that the system time  $T^s$  is put forward in function `removeOutgoingContainer(c)`, described on page 82, is not used in spite of sufficient time to perform the outgoing move of  $c_j$ , scheduled to depart before  $c_i$ .

One way to avoid such a situation is to deviate from the event structure, continuously observing approaching moves in subsequent events and, if possible, escaping the current event, perform one or several moves, and return to the remaining operations in the event in progress. This, however, will require substantial changes in the concepts of the algorithm, possibly inflicting undesirable effects on the performance. A way to handle this issue may be to allow performance of a limited number of moves “outside” the current event and possibly varying this limit. Alternatively, the issue can be dealt with by changing the initial heuristic solution and seek to shift the delayed moves towards their scheduled times by making use of crane idle time, naturally without causing other moves to be delayed. Furthermore, the technique can be embedded in the existing algorithm and run iteratively, e.g. every  $n$  moves or events, after a certain run time, etc.

Certain precautions must be taken in order not to inflict infeasibility on the solution. Shifting a delayed outgoing move by performing it before one or several moves, causing its delay, can only be allowed if the involved containers do not conflict, i.e. if they do not share positions. Otherwise, the stacking order is changed, leading to an entirely different solution. This issue, clearly possible to address, is not dealt with in this approach. Consequently, it must be checked if the move to be shifted involves any of the positions connected with the moves that it is going to precede after the shift. Furthermore, if necessary to postpone one or several moves, it must be assured that this does not cause the involved container(s) to be late for departure.

The shifting routine is implemented as described in section 3.3.2, taking an initial solution by the basic heuristic algorithm as input and, if possible, producing an alternative solution with fewer delays.

## Reducing redundant reshuffles

The third issue presented in the section introduction concerns the potential unnecessary reshuffling of containers, possibly occurring immediately after positioning or after some storage time but with no other moves performed in the meantime. An example of such a situation is the choice of a feasible high-quality position where a lower placed container is the next to be handled, causing the barely placed container to be removed from the position again. This situation may be prevented by looking ahead and avoiding selection of a position if one of the stack containers is the occasion of the subsequent event but, as previously discussed, this approach implies fundamental changes in the algorithm structure. In addition to the above obviously undesirable situation, redundant reshuffling may occur even though other moves are performed in the meantime. If intervening moves involve neither the current position of a container nor the position at which it is subsequently placed, the reshuffle is unnecessary and the two moves of the given container may be replaced by a single move, skipping the intermediate position.

Such a contraction of moves can be performed in three situations, exemplified by a container  $c$ , placed at position  $i$  and, subsequently, moved to position  $j$ , corresponding to the two moves  $m_{\rightarrow i}$  and  $m_{ij}$  respectively:

- If the storage time at position  $i$  equals 0, i.e. if container  $c$  is reshuffled immediately after being placed at  $i$ , clearly the moves  $m_{\rightarrow i}$  and  $m_{ij}$  can be contracted to one single move,  $m_{\rightarrow j}$ .
- If container  $c$  is stored at position  $i$  for some time but no other moves are performed between  $m_{\rightarrow i}$  and  $m_{ij}$ , the moves can be contracted to  $m_{\rightarrow j}$ .
- If other moves are performed between  $m_{\rightarrow i}$  and  $m_{ij}$ , contraction to the single move  $m_{\rightarrow j}$  is allowed if none of the intermediate moves involve position  $j$ .

These principles may constitute the basis for an additional improvement algorithm, running after completion of the basic heuristic. However, this work is left to future research.

## Additional interesting issues

In addition to the three main issues, discussed in above subsections, a number of other relevant aspects may be considered for future research.

As an alternative to the existing event structure, some operations - single moves or entire events - may be “conditional”, i.e. not scheduled to a certain point in time but, rather, kept in a buffer of operations that can be performed during crane idle time as long as their “expiration” time is not exceeded. This may enable sorting or preparing better stacks in time intervals where no moves are scheduled to be performed, clearly not reducing the number of moves but, possibly, improving departure time punctuality.

Rather than “filling up” the block from the arrival end and downwards, placing containers with approaching departure times further down the block, possibly by seg-

menting the block into different zones according to departure time intervals, may be advantageous.

A practical issue related to the accuracy of input data concerns the time windows, rather than points in time, within which containers are being picked up by trains or trucks for further transportation. Replacing the exact departure times by time windows, corresponding to the estimated time for a truck's arrival at the gate or the time interval spent by a train in the terminal, makes room for bringing forward or postponing outgoing moves, i.e. reducing the extend to which containers are considered early or late for departure.

A more technical issue relates to the procedure for handling events, in the present form of the basic heuristic algorithm being processed without regard for consequences for future operations. As previously discussed, looking ahead during the process of an event may prevent some disadvantageous situations such as occupying a position which is better suited for a future container, moving to a stack in which another container is close to its departure time, spending longer time than absolutely necessary to complete the event operations and, thereby, delaying subsequent events, to mention a few. Several ways to apply a look-ahead approach as well as numerous situations to anticipate can be investigated, the above examples counting only a fraction of these.

These issues are interesting aspects, technically as well as practically, and future research may show significant advances in the heuristic solution approach, presented in this chapter and tested in chapter 5. In the following sections, the two improvement routines, avoiding rejections and shifting delayed moves, are presented.

### 3.3.1 Pseudo code for procedures to avoid rejections

The procedures for avoiding potential rejections of containers is embedded in the basic heuristic algorithm and consist of function modules added to the basic heuristic function  $\text{selectMostFavourablePosition}(\mathcal{Q}^C, c)$ , displayed on page 84. The routine applies three additional parameters:  $AR$  equals 1 if not accepting rejections,  $T_c^C$  equals the time conflict between container  $c$  to be positioned and stack container  $c'$  at the position evaluated, and  $T_{p'}^C$  equals the largest time conflict represented by a container in the stack at candidate position  $p'$ . In the following,  $AR$  is assumed equal to 1.

Below, function  $\text{selectMostFavourablePositionAvoidingRejections}(\mathcal{Q}^C, c)$ , replacing function  $\text{selectMostFavourablePosition}(\mathcal{Q}^C, c)$  when  $AR = 1$ , is displayed. The extensions consist of three subroutines, the first two being part of the second feasibility check and the third performed if no feasible position is found among the candidates. For all stack containers the time conflict  $T_{c'}^C = T^S + T_{p'c'} + T_{c'}^W - D_{c'}$  is computed and if candidate position  $p'$  does not fulfil the second feasibility criterion, the position time conflict  $T_{p'}^C = \max_{c' \in \mathcal{S}_{p'}} \{T_{c'}^C\}$  is defined. In case that none of the candidate positions are feasible due to potential delays for stack containers, the position  $p'$  with the smallest time conflict  $T_{p'}^C = \min_{p'' \in \mathcal{Q}^C} \{T_{p''}^C\}$  is selected and returned as  $p^*$ .

Forcing container  $c$  to be placed at a position, possibly inflicting delays on one or several stack containers, clearly may cause undesirable situations. However, from a

**Function B\*:** selectMostFavourablePositionAvoidingRejections( $\mathcal{Q}^C, c$ )

```

 $p^* = \emptyset$ 
for all  $q_{p'} \in \mathcal{Q}^C$  do
  while  $p^* = \emptyset$  do
    if  $T^S + T_{p_c p'} + T_{p' I} \leq D_c$  then
      for all  $c' \in \mathcal{S}_{p'}$  do
         $T_{c'}^W = n \cdot T_{p'}^W + T_{p' I}$ 
         $T_{c'}^C = T^S + T_{p_c p'} + T_{c'}^W - D_{c'}$ 
        if  $T^S + T_{p_c p'} + T_{c'}^W \leq D_{c'}$  for all  $c' \in \mathcal{S}_{p'}$  then
           $p^* = p'$ 
        else
           $T_{p'}^C = \max_{c' \in \mathcal{S}_{p'}} \{T_{c'}^C\}$ 
      if  $p^* = \emptyset$  then
         $p^* = p'$  where  $T_{p'}^C = \min_{p'' \in \mathcal{Q}^C} \{T_{p''}^C\}$ 

```

practical point of view, it is required to position all containers entering the system and, moreover, due to the strict definition of the worst case moving time for stack containers  $T_{c'}^W$ , based on the extreme position worst case moving time  $T_{p'}^W$ , it is most likely that the re-computed predeparture times are scheduled significantly earlier than necessary, thus not inflicting the expected delay when actually performing the required moves.

### 3.3.2 Pseudo code for shifting routine

The shifting routine, seeking to repair an initial solution by transferring outgoing moves of delayed containers towards their scheduled departure time, is an improvement heuristic to be run after the basic heuristic algorithm for which the main loop described on page 79. Based on the initial solution provided by the greedy algorithm, allowing or not accepting rejections, the shifting routine seeks to handle containers late for departure, starting with the most delayed one, checking if its outgoing move can be shifted towards its scheduled departure time, possibly by one or several move swaps, without inflicting delays on other outgoing moves.

The shifting routine applies a number of new data structures. The stopping criterion *STOP* controls whether or not to continue the iterative process, the delay tolerance  $\mu^T$  equals the extend to which containers are accepted to be late, and the set  $\mathcal{C}^N \subseteq \mathcal{C}$  contains the containers that have not yet been treated. Equivalent to previous notation, the sets  $\mathcal{D}^O$  and  $\mathcal{D}^L$  consist of the containers being on-time and late for departure, the transportation time between two nodes,  $i$  and  $j$ , is denoted  $t_{ij}$ , and  $D_c$  denotes the scheduled departure time for container  $c$ . In addition to this,  $D_c^A$  denotes its actual departure time which may be earlier, later than, or equal to  $D_c$ .

The set  $\mathcal{M} = \mathcal{M}^F \cup \mathcal{M}^E = \{m_1, \dots, m_{\mathcal{M}}\}$  contains all moves in the current solution, clearly consisting of the subsets of full and empty moves. A given move  $m \in \mathcal{M}$  is denoted  $m_c$  if it involves container  $c_m$ , its start and end times are  $m^{start}$  and  $m^{end}$ , its

start and end positions are  $p_m^{start}$  and  $p_m^{end}$ . If  $m$  is an outgoing move, it may be denoted  $m^O$ . In the shifting algorithm, container  $c^D$  is the current most delayed container,  $p^{last}$  is its last position,  $D^{max} = D_c^A - D_c$  is the current largest delay,  $m^D = m_{c^D}^O$  is the delayed outgoing move currently considered, and  $m^{req}$  corresponds to the requested outgoing move, created by shifting the delayed outgoing move to its scheduled start time  $m^{req, start} = D_{c^D} - T_{p^{last}I}$ . Measured against the requested outgoing move  $m^{req}$ ,  $m^{pre}$  and  $m^{suc}$  denote the closest preceding and the closest succeeding full move respectively, the set  $\mathcal{M}^O$  at first contains all overlapping moves between  $m^{pre}$  and  $m^{suc}$ , and  $\mathcal{M}^{CO} \subseteq \mathcal{M}^O$  contains any conflicting overlapping moves that are not possible to swap with the latest outgoing move  $m^D$ . After adjusting  $\mathcal{M}^O$  by  $\mathcal{M}^{CO}$ ,  $\delta^F$  and  $\delta^E$  represent the total length of the overlapping full and empty moves respectively.

The shifting algorithm consists of a main loop, containing a number of separate functions. After initializing the stopping criterion  $STOP = 0$ , the delay tolerance  $\mu^T = 0$  if  $C \leq 10$  and  $\mu^T = 10$  if  $C > 10$ , the set of non-shifted containers  $\mathcal{C}^N = \mathcal{C}$ , and the largest delay  $D^{max} = 0$ , the preceding and succeeding moves  $m^{pre} = m^{suc} = \emptyset$ , the sets of overlapping and conflicting overlapping moves  $\mathcal{M}^O = \mathcal{M}^{CO} = \emptyset$ , and the lengths of overlapping full and empty moves,  $\delta^F = \delta^E = 0$ , the main loop is repeated until  $STOP = 1$ .

### Main loop: shifting routine

Below, the main loop of the shifting routine is displayed and described, and in the following sections each function is explained in detail.

#### Main loop: shifting routine

```

while  $STOP < 1$ 
   $c^D = \text{findMostDelayedContainer}(\mathcal{C}^N)$ 
  if  $D^{max} \leq \mu^T$  then
     $STOP = 1$ 
  else
     $m^{suc} = \text{findClosestSucceedingMove}(c^D)$ 
     $m^{pre} = \text{findClosestPrecedingMove}(c^D)$ 
     $\{\mathcal{M}^O, \delta^F, \delta^E = \}$   $\text{findOverlappingMoves}(c^D, m^{pre}, m^{suc})$ 
     $\text{improveSolution}(c^D, m^{pre}, m^{suc}, \mathcal{M}^O, \delta^F, \delta^E)$ 
     $\mathcal{C}^N = \mathcal{C}^N \setminus \{c^D\}$ 

```

First function  $\text{findMostDelayedContainer}(\mathcal{C}^N)$  returns the most delayed container in the current solution by searching the set of containers, not yet treated,  $\mathcal{C}^N$ . If the largest delay  $D^{max}$  does not exceed the delay tolerance  $\mu^T$ , the algorithm stops. Otherwise, the remaining four functions come into play. Functions  $\text{findClosestSucceedingMove}(c^D)$  and  $\text{findClosestPrecedingMove}(c^D)$  identify the full moves, succeeding and preceding the requested outgoing move,  $m^{suc}$  and  $m^{pre}$  respectively. Subsequently, function  $\text{findOverlappingMoves}(c^D, m^{pre}, m^{suc})$  investigates potentially overlapping moves between  $m^{pre}$  and  $m^{suc}$  and returns the set  $\mathcal{M}^O$  and the total lengths of full and empty

overlapping moves,  $\delta^F$  and  $\delta^F$ . Finally  $\text{improveSolution}(c^D, m^{pre}, m^{suc}, \mathcal{M}^O, \delta^F, \delta^F)$  performs the required feasibility checks and, if possible, improves the solution by shifting the delayed outgoing move towards the scheduled point in time. The loop is repeated after removing the treated container  $c^D$  from  $\mathcal{C}^N$ .

**Function S1: findMostDelayedContainer( $\mathcal{C}^N$ )**

The first function searches through the set of not yet treated containers  $\mathcal{C}^N$  and retrieves the most delayed container  $c^D$  by finding the largest delay  $D^{max} = \max_{c \in \mathcal{C}^N} \{D_c^A - D_c\}$ . The pseudo code for the function is displayed in the box below and an overview of the procedure is provided in figure 3.7.

**Function S1: findMostDelayedContainer( $\mathcal{C}^N$ )**

```

for all  $c \in \mathcal{C}^N$  do
  if  $D_c^A - D_c > D^{max}$  then
     $D^{max} = D_c^A - D_c$ 
     $c^D = c$ 

```

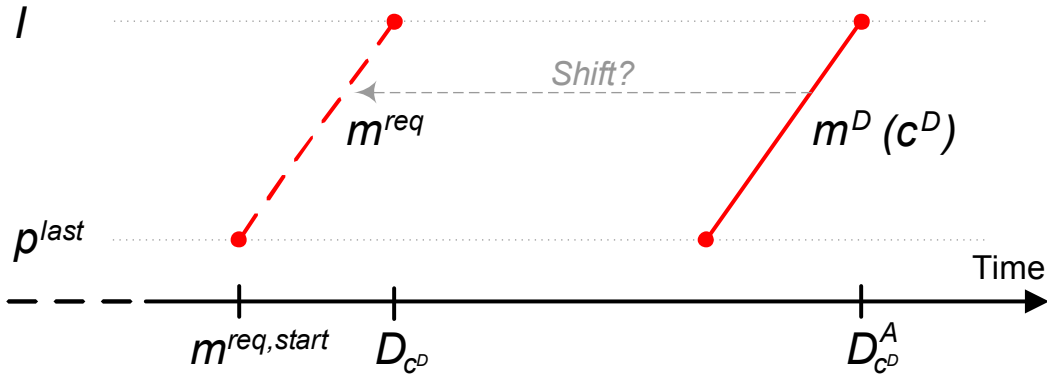


Figure 3.7: Overview of the first function  $\text{findMostDelayedContainer}(\mathcal{C}^N)$ , finding the most delayed container  $c^D$ , its latest outgoing move  $m^D$ , depicted by a solid red line, and its last position  $p^{last}$ . The dashed red line illustrates the requested outgoing move  $m^{req}$  which is the desired results of the potential shift of  $m^D$ .

**Function S2: findClosestSucceedingMove( $c^D$ )**

Function  $\text{findClosestSucceedingMove}(c^D)$  searches through the set of full moves  $\mathcal{M}^F$ , identifying the move which immediately succeeds the requested outgoing move  $m^{req}$ . Starting from the last move in the sequence  $m_{\mathcal{M}}$ , the function checks if the current move starts after the potential empty move from the departure place to its start position, if the current move is performed before the previously found closest succeeding move, and that it is not equal to the latest outgoing move  $m^D$ . If these conditions are fulfilled a

second check ensures that there is no position overlap between the current move and the last position for the most delayed container  $p^{last}$  before setting the closest succeeding move  $m^{suc}$  equal to the current move. If there is a position overlap but the current move is performed after the latest outgoing move,  $m^{suc}$  is also set to the current move as shifting  $m^D$  will not imply a swap of any kind in that situation, i.e. the position overlap is irrelevant. The below box displays the function pseudo code and 3.8 illustrates the procedure.

**Function S2:** findClosestSucceedingMove( $c^D$ )

```

let  $m^{suc} = m_{\mathcal{M}}$ 
for all  $m \in \mathcal{M}^F$  where  $m = m_{\mathcal{M}}..m_1$  do
  if  $m^{start} \geq D_{c^D} - T_{I_{p_m^{start}}} \wedge m^{start} < m^{suc,start} \wedge m \neq m^D$  then
    if  $p_{m^D}^{start} \neq \{p_m^{start}, p_m^{end}\} \vee m^{start} > m^{D,end}$  then
       $m^{suc} = m$ 

```

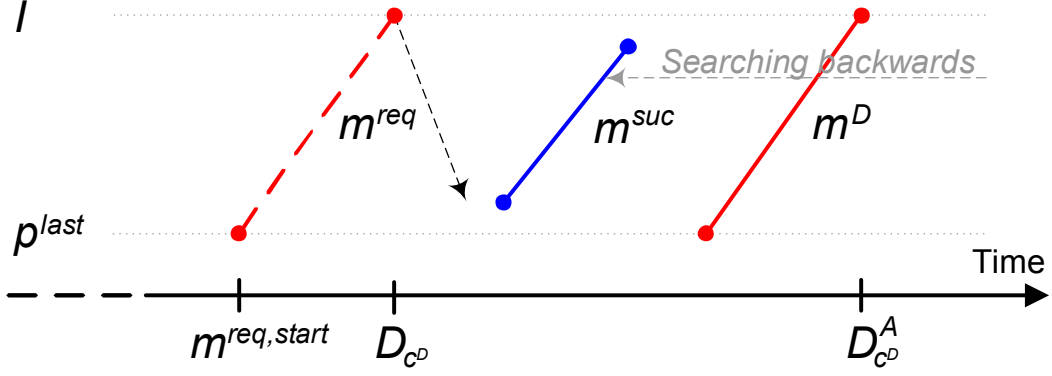


Figure 3.8: Overview of the second function findClosestSucceedingMove( $c^D$ ), finding the closest succeeding move  $m^{suc}$  by searching backwards through all full moves  $\mathcal{M}^F$  until meeting the end time for the required empty move, depicted by a dashed arrow, from the requested outgoing move  $m^{req}$ .

**Function S3:** findClosestPrecedingMove( $c^D$ )

Function findClosestPrecedingMove( $c^D$ ) resembles findClosestSucceedingMove( $c^D$ ), identifying the closest preceding move  $m^{pre}$ , considering the latest outgoing move if performed at its scheduled time,  $m^{req,start}$ . Now searching the set of full moves  $\mathcal{M}^F$  from the first move in the sequence  $m_1$  to the closest succeeding move  $m^{suc}$ , it is checked if the current move's end time plus the time, required to perform the empty move from its end position to the most delayed container's last position, does not exceed the scheduled start time for the latest outgoing move and if the current move is performed after the previously found closest preceding move, in which case  $m^{pre}$  is set equal to the current

move. The pseudo code is displayed below and figure 3.9 provides an overview of the function procedure.

**Function S3:** findClosestPrecedingMove( $c^D$ )

```

let  $m^{pre} = m_1$ 
for all  $m \in \mathcal{M}^F$  where  $m = m_1..m^{suc}-1$  do
  if  $m^{end} + T_{p_m^{end}p^{last}} \leq m^{req,start} \wedge m^{start} > m^{pre,start}$  then
     $m^{pre} = m$ 

```

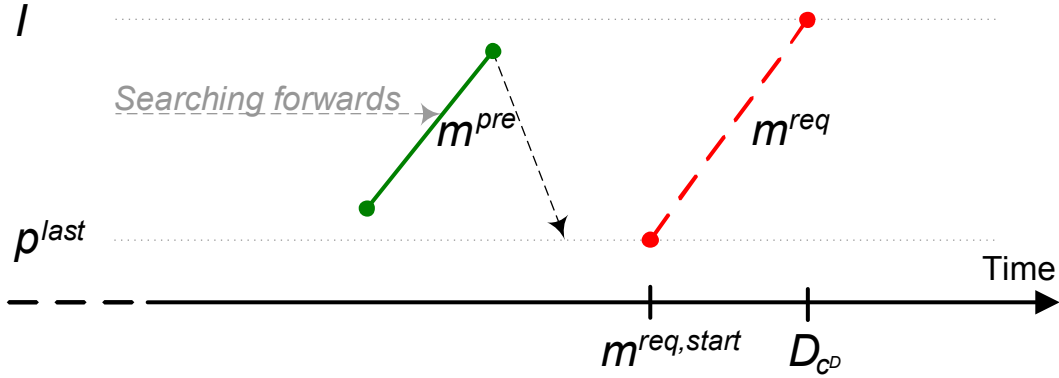


Figure 3.9: Overview of the third function findClosestPrecedingMove( $c^D$ ), finding the closest preceding move  $m^{pre}$  by searching forwards through all full moves  $\mathcal{M}^F$  until meeting the start time for the required empty move, depicted by a dashed arrow, to the requested outgoing move  $m^{req}$ .

**Function S4:** findOverlappingMoves( $c^D, m^{pre}, m^{suc}$ )

Based on the closest preceding and succeeding full moves,  $m^{pre}$  and  $m^{suc}$ , function findOverlappingMoves( $c^D, m^{pre}, m^{suc}$ ) searches for potential overlapping moves in the time interval between them, the purpose being to identify moves that may or may not be postponed if performing the latest outgoing move at its scheduled time.

After initializing  $\mathcal{M}^O$  and  $\mathcal{M}^{CO}$  to  $\emptyset$ , all full moves between  $m^{pre}$  and  $m^{suc}$  are at first added to  $\mathcal{M}^O$  unless equal to the latest outgoing move. Subsequently, for each overlapping move it is checked if the current move is an on-time or late outgoing move and, if so, it is added to the set of conflicting overlapping moves  $\mathcal{M}^{CO}$  that are not allowed to be postponed by the shifting procedure due to the delay being inflicted. Furthermore, if the current move involves the most delayed container's last position the move is added to  $\mathcal{M}^{CO}$  as the position conflict makes the potential move swap infeasible. When including the current move in  $\mathcal{M}^{CO}$  the set of overlapping moves  $\mathcal{M}^O$  is reset as the latest outgoing move cannot be shifted to earlier than the largest element in  $\mathcal{M}^{CO}$  due to the above conflicts.

If conflicting overlapping moves exist, the closest preceding move  $m^{pre}$  is updated to the last move  $m_{\mathcal{M}^{CO}}$  in  $\mathcal{M}^{CO}$  as the latest outgoing move, if shifted, must be performed



after the last conflicting move and before the closest succeeding move. Finally, if one or several overlapping moves exist, the total lengths of the overlapping full and empty moves,  $\delta^F$  and  $\delta^E$ , are computed. The box below disposes the function pseudo code and figure 3.10 illustrates the principle in the procedures.

**Function S4:** findOverlappingMoves( $c^D, m^{pre}, m^{suc}$ )

```

let  $\mathcal{M}^O = \mathcal{M}^{CO} = \emptyset$ 
for all  $m \in \mathcal{M}^F$  where  $m = m^{pre} + 1 .. m^{suc} - 1$  do
  if  $m \neq m^D$  then
     $\mathcal{M}^O = \mathcal{M}^O \cup \{m\}$ 
    if  $c_m \in \mathcal{D}^O \cup \mathcal{D}^L \vee p_m^{start} = \{p_m^{start}, p_m^{end}\}$  then
       $\mathcal{M}^{CO} = \mathcal{M}^{CO} \cup \{m\}$ 
     $\mathcal{M}^O = \emptyset$ 
  if  $\mathcal{M}^{CO} \neq \emptyset$  then
     $m^{pre} = m_{\mathcal{M}^{CO}}$ 
  if  $\mathcal{M}^O \neq \emptyset$  then
     $\delta^F = \sum_{m \in \mathcal{M}^O} T_{p_m^{start} p_m^{end}}$ 
     $\delta^E = \sum_{m \in \mathcal{M}^O} T_{p_m^{end} p_{m+1}^{start}}$ 

```

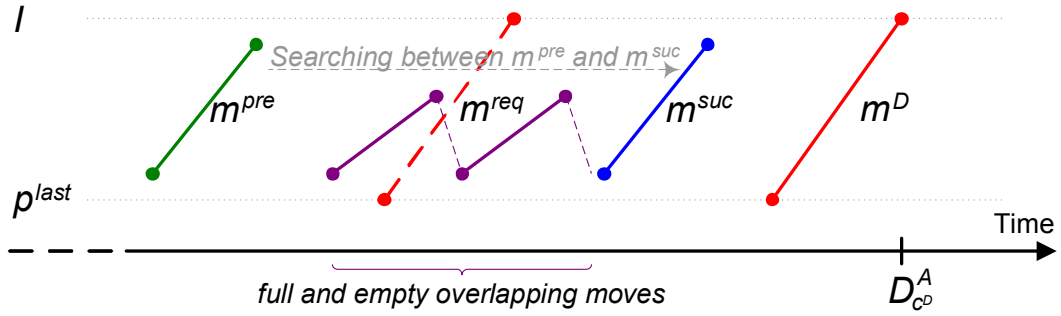


Figure 3.10: Overview of the fourth function findOverlappingMoves( $c^D, m^{pre}, m^{suc}$ ), searching for overlapping moves, depicted by purple lines, between the closest preceding move  $m^{pre}$  and the closest succeeding move  $m^{suc}$ .

**Function S5:** improveSolution( $c^D, m^{pre}, m^{suc}, \mathcal{M}^O, \delta^F, \delta^E$ )

Knowing the closest preceding move  $m^{pre}$ , the closest succeeding move  $m^{suc}$ , and possibly overlapping full moves  $\mathcal{M}^O$ , function improveSolution( $c^D, m^{pre}, m^{suc}, \mathcal{M}^O, \delta^F, \delta^E$ ) seeks to repair the initial solution by shifting the latest outgoing move towards its scheduled time. Three situations, two of them special cases, can occur:

- 1 The closest succeeding move  $m^{suc}$  equals the latest outgoing move  $m^D$ , i.e. both equal to the last move in the sequence of full moves  $\mathcal{M}^F$ .

- 2 The closest succeeding move  $m^{suc}$  is performed after the latest outgoing move  $m^D$ , i.e. shifting  $m^D$  does not imply any moves swaps.
- 3 The closest succeeding move  $m^{suc}$  is performed before the latest outgoing move  $m^D$ , i.e. shifting  $m^D$  implies one or several moves swaps.

In cases 1 and 2, the same procedure is executed. If the end time of the required empty move from the closest preceding move  $m^{pre}$  to the latest container's last position  $p^{last}$  is earlier than the scheduled start time of the shifted outgoing move  $m^{req}$ , it is simply shifted to the requested time  $m^{req,start}$ . Otherwise, i.e. if the preceding empty move ends at or after the scheduled start time,  $m^D$  is shifted to immediately after the preceding empty move, thus possibly remaining late.

In case of the closest succeeding move  $m^{suc}$  being performed before the latest outgoing move  $m^D$ , as will often be the situation, it is checked if the shifted move can be performed before the overlapping moves without delaying the closest succeeding move, i.e. if summing the end time for the empty move between the closest preceding move and the requested outgoing move, the length of requested outgoing move, and the total lengths of overlapping full and empty moves does not exceed the start time for the closest succeeding move. If so, shifting the latest outgoing move is feasible and can be performed, i.e.  $m^D$  can be transferred either to the requested time or to immediately after the preceding empty move, equivalently to cases 1 and 2 above, and if overlapping moves exist they can be postponed by the required time intervals, thus resulting in an improved solution. The box below presents the pseudo code for the function.

**Function S5:** improveSolution( $c^D, m^{pre}, m^{suc}, \mathcal{M}^O, \delta^F, \delta^E$ )

```

if  $m^{suc} = m^D \vee m^{suc,start} > m^{D,end}$  then
  if  $m^{pre,end} + T_{p_{m^{pre}}^{end} p^{last}}^{end} < m^{req,start}$  then
    shift  $m^D$  to  $m^{req,start}$ 
  else
    shift  $m^D$  to  $m^{pre,end} + T_{p_{m^{pre}}^{end} p^{last}}^{end}$ 
if  $m^{suc,end} > m^{D,start}$  then
  if  $m^{pre,end} + T_{p_{m^{pre}}^{end} p^{last}}^{end} + T_{p^{last} I} + \delta^F + \delta^E \leq m^{suc,start}$  then
    if  $D_{c^D} + \delta^F + \delta^E \leq m^{suc,start}$  then
      shift  $m^D$  to  $m^{req,start}$ 
    if  $\mathcal{M}^O \neq \emptyset$  then
      postpone all  $m \in \mathcal{M}^O$ 
  else
    shift  $m^D$  to  $m^{pre,end} + T_{p_{m^{pre}}^{end} p^{last}}^{end}$ 
    if  $\mathcal{M}^O \neq \emptyset$  then
      postpone all  $m \in \mathcal{M}^O$ 

```

### 3.4 Conclusive remarks

The heuristic solution approach presented in this chapter consists of a greedy construction algorithm and an improvement routine, seeking to repair the initial heuristic solution.

The basic heuristic algorithm builds a solution by placing each arrival container, one at a time, by considering possible future operations but without using information about containers, arriving after the current one. Thus, the greedy algorithm constitutes an on-line approach, handling one task at a time without drawing on information about future events. However, implementing look-ahead elements, such as considering e.g. a certain number of containers to arrive in the future, may improve the proposed algorithm.

Perspectives for further development of the heuristic may also include reading an initial block layout into the algorithm, thus building the solution on an existing situation rather than starting from an empty block, alternative repair functions in addition to - or replacing - the presented shifting routine, and investigation of the effect of reshuffling and reorganizing stacks, not only when forced by lower placed containers but also if it seems advantageous for future operations. More technical issues to investigate may include possible time-savings by insertion of events instead of sorting the entire list of events in every main loop iteration, applying different functions to shift late outgoing moves, etc.

The shifting routine is constructed to attempt to transfer a delayed container's outgoing move to a point in time close to its scheduled departure time, if possible, performing the move between the closest preceding and succeeding moves. The routine considers postponing potentially overlapping moves between these but not changing the closest succeeding move even though this may be possible. Furthermore, if the requested shift is not feasible, no improvement of the given container's actual departure time is performed.

Alternatively, stepwise shifting a delayed outgoing move forwards by iteratively performing swaps with moves that may be postponed, i.e. checking that no delays are inflicted and that no position conflicts exist, may open up the possibility of reducing delays to some extent instead of abandoning all plans of improvement if the full shift cannot be performed.

The heuristic algorithm, as an alternative to the implementation of the models in standard algebraic languages, may prove to be a very efficient solution approach due to the construction design, requiring very short computation time. However, this depends on the performance of the algorithm concerning solution quality compared to run times. In the following chapter, presenting the computational results of this research, the heuristic approach is compared to the models' performance and conclusive remarks on the different solution approaches are stated.

## Chapter 4

# Computational platforms and test problems

This chapter describes the computational platforms for implementation of the solution procedures and the test problems that are used to produce the computational results, presented in chapter 5.

### 4.1 Computational platforms

Three different sets of implementation tools, modeling languages and/or solvers have been used in this study. The container terminal problem (CTP) model is implemented in GAMS and solved by the Cplex MIP solver. The two CPP models are implemented in Xpress-Mosel and solved by the Xpress-optimizer. The heuristic is implemented in Java using the Eclipse environment.

GAMS is a general algebraic modeling system developed for implementing mathematical optimization models. It interacts with a number of solvers, including ILOG's CPLEX MIP optimizer [18, 23]. Dash Optimization's Xpress-Mosel is a development tool similar to GAMS, constructed for implementing mathematical optimization models. Xpress-Mosel is somewhat more accessible than GAMS, but in both systems, models are easily implemented more or less directly from their mathematical form. Whereas implementation in GAMS is done in a standard text editor - Emacs in this study - Mosel is part of the Xpress-MP product, also including the Windows-based Xpress-IVE environment in which implementation, compiling, execution, and solution analysis can be performed, and a number of Xpress-optimizers [13]. Both the CPLEX and Xpress MIP solvers used in this study apply branch and cut algorithms.

Java is a standard programming language, developed by Sun Microsystems as an open source product, that includes a wide range of features, facilitating the algorithm implementation [54]. Eclipse, also an open source tool, is a visual implementation environment, including several features for developing projects, compiling and debugging, controlling runs etc. [16].

## 4.2 Data and test problems

This section presents the different sizes, types, and classes of test problems and real-life problem instances which are used to validate, test, and compare the models and the heuristic algorithm. Part of the problem instances are generated artificially and part of them consist of real-life data from a medium-size European port container terminal. All data files can be found on the enclosed CD-rom in appendix B.

### 4.2.1 Problem types

Three types of problem structures represent different arrival and departure time patterns for the containers: Stair, cone, and mix structures. In the stair test problems, containers depart from the block in the order in which they arrive. The cone test problems form a pyramid with containers departing from the block in the reverse order of the arrival sequence. The stair structure is likely to imply more reshuffling and/or require a larger number of positions than the cone structure as it is not possible to stack the containers as they arrive and remove them in the reverse sequence, emptying the stack from top to bottom. Thus, the stair structure represents the hardest problem type in a family of instances whereas the cone structure represents the easiest case. The mix test problems represent neither pure stair nor cone structures but a mix of the different arrival and departure time patterns. In mix instances, clearly, both stair and cone structures will occur and, obviously, the containers in the smaller subcones may be advantageous to stack on top of each other. In practice, the mix structure is by far the most realistic. Figure 4.1 illustrates the three different types of test problem structures, exemplified by three containers.

While the structure of the stair and cone instances can be defined unambiguously, given the size of the problem, there are several instances of the mix problem types. Illustrated with three containers  $c = 1, 2, 3$ , arriving in that order, i.e.  $A_1 < A_2 < A_3$ , the departure times for the stair and cone problems will be arranged as  $D_1 < D_2 < D_3$  and  $D_3 < D_2 < D_1$  respectively. The complete set of mix instances represent all possible combinations of departure times except the two cases described above.

Given the the number of containers  $C$  and a sequenced list of arrival times  $A_1 < A_2 < \dots < A_C$ , all possible combinations of departure times sequences are  $C!$  of which the two instances of strictly ascending and descending departure times represent the stair and cone structures. However, only in the very small sets of test problems all mix instances are included.

### 4.2.2 Problem classes

Three classes of problem instances are introduced: Small-scale, medium-scale, and large-scale problems, representing different features and purposes. Each of the classes consists of a number of subclasses defined by the number of containers  $C$ . Each test problem instance is represented in a tight and a scaled version, corresponding to the basic instance, being infeasible if reducing the time span marginally, and a version with an extended

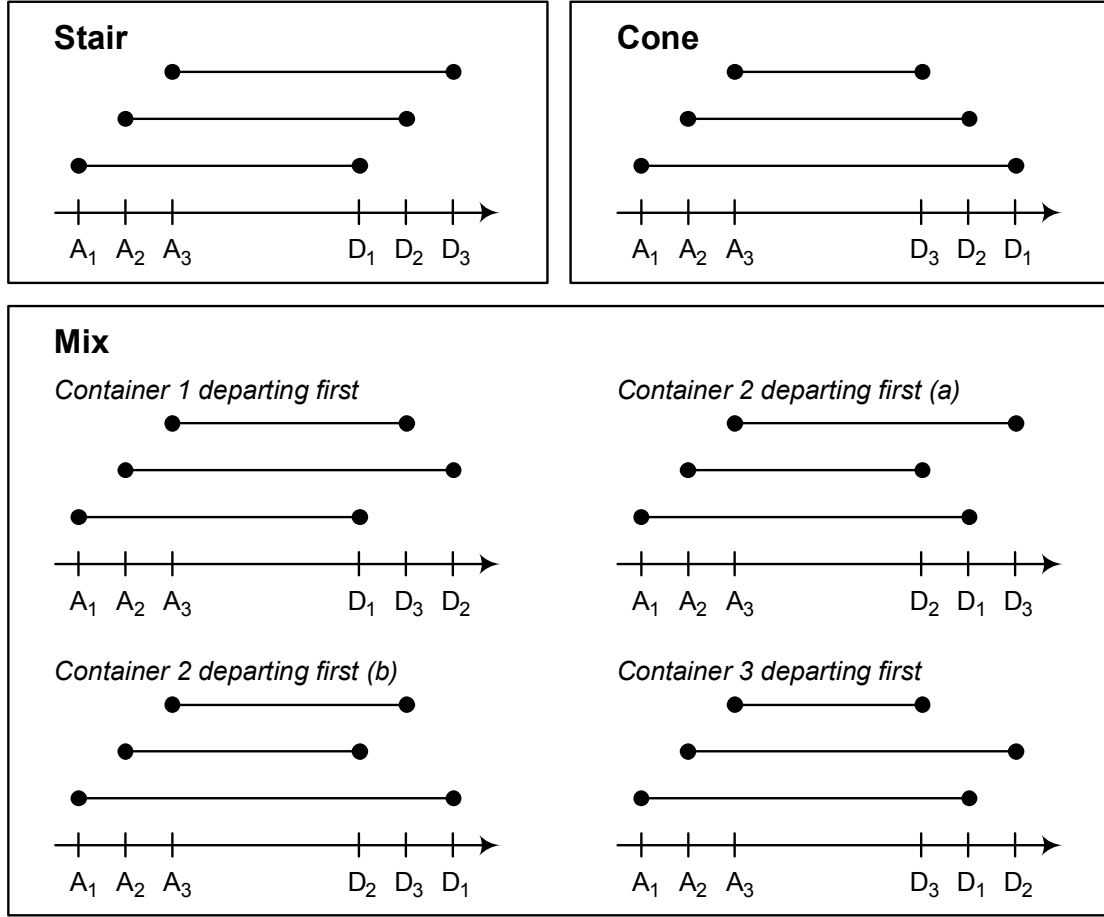


Figure 4.1: Schematic overview of the three types of test problem structures: Stair, cone, and mix. The departure sequence of the stair and cone instances can be defined unambiguously whereas the mix structure represents several problem instances, depending on the sequence of departure times.

time dimension, resulting in more scattered arrival and departure times. Naturally, none of the large-scale class instances are the object of variation of the time dimension.

It should be noted that a modeling difference between the CPP and the CPPT model implies that a given tight instance for the CPP model must be extended with one time unit in order to be feasible in the CPPT model. The explanation for this is the time-discretized structure of the CPPT model, requiring one time slot for the arrival time as well as the departure time, as opposed to the two points in time in the CPP model. The difference is illustrated in figure 4.2 by an example of a container  $c$  with  $A_c = 1$  and  $D_c = N$ , implying that the container must be at the arrival place in time slot 1 and at the departure place in time slot  $N + 1$  in the CPPT model.

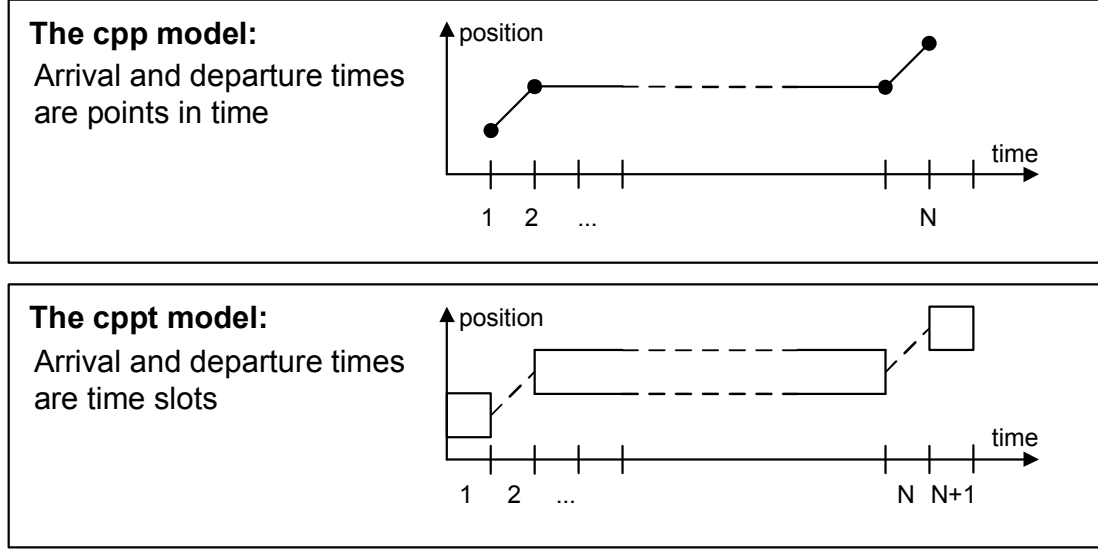


Figure 4.2: Illustration of the time dimension modeling difference between the CPP and the CPPT model. As the CPPT model is based on time slots, arriving in time slot 1 corresponds to an arrival time of 1 and departing in time slot  $N + 1$  corresponds to a departure time of  $N$ . Therefore, to represent the same problem instance, an extra time slot must be added to ensure feasibility of the CPPT model.

### The small-scale class

The small-scale problem class serves the purpose of validating the models and the heuristic as well as tuning the heuristic parameters. Subclasses contain instances with 3 - 5 containers, generated by hand or semi-automatically. The small-scale test problems are quite transparent and the solutions are easily appreciated for which reason they are well-suited for demonstrating features and properties of the problems as well as the models and the heuristic.

Each subclass consists of one stair, one cone, and  $2C$  mix instances, resulting in  $\sum_{C=3}^5 (2C + 2) = 30$  tight small-scale problem instances. Consequently, also 30 scaled instances belong to this class. Problem instance files can be found in appendix B.

The instances in each subclass resemble each other as regards the number of positions  $P$ , time dimension  $L$ , and stacking height  $H$ . Therefore, the heuristic penalty parameters, depending on these system characteristics, are computed once for all subclass instances by the largest value of  $dist$ ,  $time$ , and  $stack$ , see section 3.1.3 on page 68 for an explanation of the procedure.

In the 3 container subclass,  $P = 2$  and  $H = 2$  in all instances whereas  $L$  varies between 10 and 12 in the tight and 30 and 36 in the scaled test problems. Based on the worst case outcomes,  $dist^W = R + B - 1 = 2$ ,  $time^W = L^{max} = \{12, 36\}$ , and  $stack^W = H = 2$ , the default values of the tuning parameters  $(\rho, \mu, \alpha, \sigma) = (2, 1, 0.144, 0.116, 0.024, 0.192, 0.144, 1.0, 1.0, 1.0, 1.0)$  and  $(2, 2, 0.144, 0.039, 0.008, 0.064, 0.144,$

1.0, 1.0, 1.0, 1.0) for the tight and scaled instances respectively.

The same procedures are used to compute the penalty parameters for the 4 containers and 5 containers subclass instances with  $(P, L^{max}, H) = (2, \{18, 54\}, 3)$  and  $(3, \{22, 66\}, 3)$  respectively, resulting in  $(\rho, \mu, \underline{\alpha}, \underline{\sigma}) = (2, 1, 0.144, 0.077, 0.016, 0.128, 0.096, 1.0, 1.0, 1.0, 1.0)$  and  $(2, 2, 0.144, 0.026, 0.005, 0.043, 0.096, 1.0, 1.0, 1.0, 1.0)$  for the  $C = 4$  and  $(3, 1, 0.096, 0.063, 0.013, 0.105, 0.096, 1.0, 1.0, 1.0, 1.0)$  and  $(3, 2, 0.096, 0.021, 0.004, 0.035, 0.096, 1.0, 1.0, 1.0, 1.0)$  for  $C = 5$ .

In all small-scale test problems, the search range  $\rho$  is set to  $P$ , the small time interval  $\mu$  is set to 3% of  $L$ , and the crane velocity  $V$  is set to 1 which is far slower than in reality but provides a transparency of the solutions. Configuration files can be found in appendix B.

### The medium-scale class

The medium-scale problem class is used for performance test of the models and comparison with heuristic solutions. Furthermore, results from the medium-scale test phase serve as a basis for determination of the heuristic tuning parameters. The subclasses consist of 6, 8, and 10 container instances, generated semi-automatically or automatically. Though the problems are still very small compared to real-life data, model sizes grow significantly with the number of containers and the medium-scale test problems, therefore, qualify for identifying limitations of the models and performing more extensive tests of the solution approaches than is possible with the small-scale class instances.

Each subclass consists of one stair, one cone, and  $C$  mix instances, resulting in 30 tight and 30 scaled instances in the medium-scale problem class. Problem instance files can be found in appendix B.

Equivalently to the small-scale test phase, the heuristic parameters for the first tuning assume their default values. The system characteristics  $(P, L^{max}, H)$ , equal to  $(3, \{28, 84\}, 3)$ ,  $(3, \{39, 117\}, 3)$ , and  $(4, \{78, 234\}, 3)$  for the 6, 8, and 10 containers subclasses respectively, result in the parameter values  $(\rho, \mu, \underline{\alpha}, \underline{\sigma}) = (3, 1, 0.096, 0.050, 0.010, 0.082, 0.096, 1.0, 1.0, 1.0, 1.0)$  and  $(3, 3, 0.096, 0.017, 0.003, 0.027, 0.096, 1.0, 1.0, 1.0, 1.0)$  for  $C = 6$ ,  $(4, 2, 0.096, 0.036, 0.007, 0.059, 0.096, 1.0, 1.0, 1.0, 1.0)$  and  $(4, 4, 0.096, 0.012, 0.002, 0.020, 0.096, 1.0, 1.0, 1.0, 1.0)$  for  $C = 8$ , and  $(4, 3, 0.096, 0.018, 0.004, 0.030, 0.072, 1.0, 1.0, 1.0, 1.0)$  and  $(4, 8, 0.096, 0.006, 0.001, 0.010, 0.072, 1.0, 1.0, 1.0, 1.0)$  for  $C = 10$ . As well as in the small-scale test phase,  $\rho = P$ ,  $\mu = 3L/100$ , and  $V = 1$ . Configuration files can be found in appendix B.

### The large-scale class

The large-scale problem class is used to test the heuristic. Instances in the subclasses consist several hundreds or thousands of containers, making them impossible to solve by the models. The problems represent real-life data from Oslo Container Terminal (OCT), the size of an instance corresponding to the number of containers included. Tests of the large-scale class problems enable thorough investigation of properties and performance of the heuristic. While the small- and medium-scale class problems represent both stair,



cone, and mix instances, only the latter structure occurs in the large-scale test problems since, practically, no real-life data is arranged in a strict stair or cone structure.

The large-scale instances are generated based on a large data set consisting of a total of 2245 containers, arriving during the period May - November 2005 and departing in November 2005. The instances are categorized in four subclasses: 1 week, 2 weeks, 3 weeks, and 1 month - denoted 4 weeks - data sets, corresponding to containers departing in these intervals. Prior to the specified time period, a warm-up phase, consisting of containers arriving during the preceding six months, constitutes the building up of the block layout. This somewhat corresponds to reading a block layout into the algorithm before solving the instance of “actual” containers.

In the 1 week instances, containers arrive in the period from May 6 through November 7 and depart between November 1 and November 7. 357 containers arrive before the first week of november and 499 arrive during the week. In the 2 weeks, 3 weeks, and 4 weeks instances, as well as in the 1 week subclass, 357 containers arrive during the warm-up period from May 6 to October 31 and respectively 999, 1503, and 1888 containers arrive and depart during the first 2, 3, and 4 weeks in November.

Each subclass consist of a number of problem instances of size  $C = 856$ ,  $C = 1356$ ,  $C = 1860$ , and  $C = 2245$  respectively, differing in planning strategies by variation of the number of positions available and the search range  $\rho = \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{4}, \frac{3}{4}, 1\}P$ . Besides testing the consequence of reserving a different number of positions for the containers - corresponding to varying the block size - the proportion between the number of rows and bays is represented in the different instances in each subclass. With  $(R, B, P) = (7, 14, 98)$ ,  $(7, 18, 126)$ ,  $(7, 23, 161)$ ,  $(8, 16, 128)$ ,  $(8, 20, 160)$ , and  $(8, 27, 216)$ , representing  $R/B$  ratios of 50%, 40%, and 30% and load factors varying between 3.96% and 22.91%, each subclass contains 36 problem instances. With four subclasses and two different search strategies investigated, the large-scale class consists of a total of 288 problem instances. Problem instance files can be found in appendix B.

In all large-scale subclasses, the stacking height  $H$  is set to 5, the small time interval  $\mu$  is set to the cut-off value 120, corresponding to allowing two hours early departures, and the crane velocity  $V$  is set to 25. Measuring in rows/bays per minute, this corresponds to an average speed of 10 km/h which conforms to real-life conditions. Whereas the scaling factors ( $\sigma^{NE}$ ,  $\sigma^{SE}$ ,  $\sigma^{NW}$ , and  $\sigma^{SW}$ ) are kept at 1.0 in the small- and medium-scale class instances, two different search strategies - a combined forward and backward search and a strict forward search - are tested for the large-scale class problems, corresponding to  $\underline{\sigma} = (1.0, 1.0, 1.0, 1.0)$  and  $\underline{\sigma} = (1.0, 1.0, 0.0, 0.0)$  respectively.

With the system characteristics  $P = \{98, 126, 161, 128, 160, 216\}$ ,  $L^{max} = 299, 460$ , and  $H = 5$  in all subclasses, the heuristic penalty parameters assume the default values  $\alpha^{dist} = (0.014, 0.012, 0.010, 0.013, 0.011, 0.008)$ ,  $\alpha_k^{time} = 0.000$  for all  $k$ , and  $\alpha^{stack} = 0.058$ . Due to the very small  $\alpha_k^{time}$  values, cut-off values of 0.001 for  $\alpha_0^{time}$  and  $\alpha_2^{time}$  and 0.01 for  $\alpha_3^{time}$  are applied in the first tuning phase. Configuration files can be found in appendix B.

### 4.2.3 Naming conventions

Small- and medium-scale test cases are named by the subclass, the type, and whether it is a tight or a scaled instance. The subclass is defined by the number of containers  $C$ , the problem type is denoted by **S** for stair, **C** for cone, and **M** for mix, and **t** and **s** signify tight and scaled instances respectively. In the mix problems, a number is appended to the name, indicating the first departing container in the given instance. In the small-scale class, two mix instances with each container departing first are included, thus being denoted **A** and **B** respectively. Examples of small-scale instance names are **3S.t** for the 3 containers stair instance in the tight version, **4C.s** for the scaled 4 containers cone instance, **5M1A.t** and **5M3B.s** for a tight 5 containers mix instance with container 1 departing first and a scaled 5 containers mix instance with container 3 departing first.

In the large-scale class, instances are named according to the subclass, the number of positions, and the block proportions represented by the  $R/B$  ratio, succeeded by **f** if applying a strict forward search strategy. Thus, examples of large-scale instance names are **856.98.50** for a 1 week instance with 98 positions, twice as many bays as rows, and a combined forward and backward search strategy, **1356.126.40.f** for a 2 weeks instance with 126 positions,  $R/B = 0.4$ , and a strict forward search strategy, and **2245.216.30** for a 4 weeks instance with 216 positions and an oblong block format.

In the following chapter, the computational results for the instances, described in this section, are reported and discussed.



## Chapter 5

# Computational results

This chapter presents the computational results of the research concerning the container positioning problem (CPP). Following an introduction to the solution process, section 5.1 concerns the computational results for the CPP and CPPT models and section 5.2 provides the results for the heuristic. A comparison of the solution approaches and some conclusive remarks are presented in section 5.3.

The main solution process consists of three phases: 1) Solving the small-scale problems in order to validate both models and the heuristic and also to perform a preliminary investigation of the heuristic tuning parameters, 2) solving the medium-scale problems, partly to test the performance of the CPP and CPPT model as well as comparing with the heuristic results and partly to determine good values of the heuristic tuning parameters, and 3) solving the large-scale problems by the heuristic, building on the experience from the small- and medium-scaled tuning results, in order to test the algorithm performance on real-life data sets, applying different planning strategies. In section 4.2 an overview of the problem instances which are used in the computational study presented in this chapter is provided. In addition to phases 1 and 2, model re-runs on selected time-consuming instances are performed with an extended run time limit and an investigation of the potential gain of relaxing the lifo constraints is carried out.

In the small- and medium-scale test phases, besides validating the approaches, solution of tight and scaled versions of the problem instances serves the purpose of comparing model solutions to heuristic results. A comparison between the CPPT model and the heuristic is made by excluding empty moves in the heuristic procedure by which restrictions, equivalent to the CPPT model, except for the strict arrival and departure time constraints, are obtained. (We recall that the CPP model differ from the CPPT model and the heuristic by not including capacity constraints, preventing simultaneous moves of containers and violation of the maximum stacking height.) The scaled versions of the test problems mainly serve as a comparison of the run time of the different approaches.

In the large-scale test phase, due to extensive time-consumption when solving the models, no comparison between model and heuristic results is made. The computational results from this phase are only related to performance of the heuristic when treating large-scale real-life problem instances.

## 5.1 Results for the CPP and CPPT models

In this section, the results from the solution process for the CPP and CPPT models are documented. The preliminary CTP model, serving as a first conceptual modeling approach, capturing the entire port container terminal, is not treated further than the initial validation, presented in section 2.1.

The main solution procedure includes validation of the models and evaluation of their efficiency and performance. A total of 120 problem instances are solved by each model: 30 tight and 30 scaled instances in both the small- and medium-scale class. The CPP model results are presented in table 5.2 - 5.5 and each solution is documented in appendix D. Likewise, the results for the CPPT model are stated in the four tables 5.6 - 5.9 and appendix E contains the solutions for each of the 120 instances.

A limit on the run time of 1,800 seconds from solving the LP relaxation in the root node is set, assuming that 30 minutes is a reasonable time interval to plan the container handling prior to a vessel's arrival in the port terminal. To investigate the gain of spending more time on finding better solutions, re-runs of a significantly longer run time are performed for some of the problems, where optimum is not reached within the 1,800 second's time limit. The results of these re-runs are displayed separately in table 5.10 in section 5.1.3 and the solutions can be found in appendix F.

In addition to the small- and medium-scale test phases and the re-runs of some of the highly time-consuming instances, the potential gain of relaxing the lifo constraints is investigated. The results of these tests are represented in tables 5.11 and 5.12 in section 5.1.4, the solutions in appendix G.

In the solution tables 5.2 - 5.9, each row represents a problem instance, denoted by the subclass  $C = 3, 4$ , and  $5$  in the small-scale class and  $C = 6, 8$ , and  $10$  in the medium-scale class, the type, S for stair, C for cone, and M1, M2, ..., MC for mix, and t or s for tight or scaled versions respectively.

Three **Solution** columns state the objective value OPT, the total number of moves to positions #M and the gap between the global lower and upper bound, GLB and GUB, at termination GAP. We recall that the objective value of both models is the sum of the total transportation time and the total number of positionings, i.e. to compare to the the transportation time and the number of moves in the heuristic, #M must be subtracted from OPT and  $C$  must be added to #M. If the gap is not closed, the OPT column states [GLB,GUB].

Next, four **Root** columns display the characteristics of the root node. The lower and upper bound in the root node, corresponding to the the best LP bound and the current best feasible solution, if existing, and the gap between these are denoted RLB, RUB, and GAP. The quality of the root node lower bound, measured in percent of the global upper bound, is denoted QUAL. If no feasible solution is found in the root node, i.e.  $RUB = \infty$ , - is displayed in the GAP column.

Finally, the **Run** columns document the number of nodes in the branch and bound tree #N, and CPU<sup>1</sup> and CPU\* state the elapsed time in seconds to find the first optimal solution and to close the gap between GLB and GUB respectively. A CPU time of 0.0

seconds corresponds to a run time under 0.1 seconds. If optimum is not reached within the time limit, CPU<sup>1</sup> states the time for the best solution found (- if none) and - is stated in the CPU\* column.

Keys to the solution tables are summarized in table 5.1, also explaining the computation of the column entries.

| <b>Solution</b>  |                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPT              | Objective value of an optimal solution, $OPT = GLB$ (global lower bound) = $GUB$ (global upper bound). If no optimal solution is found, $[GLB, GUB]$ is stated.                                                                                 |
| #M               | Number of moves to positions in an optimal solution. If no optimal solution is found, - is stated. The difference between the #M and OPT is the total transportation time.                                                                      |
| GAP              | Optimality gap in percent, $GAP = (GUB - GLB)/GLB \cdot 100$ .                                                                                                                                                                                  |
| <b>Root</b>      |                                                                                                                                                                                                                                                 |
| RLB              | Root node lower bound.                                                                                                                                                                                                                          |
| RUB              | Root node upper bound.                                                                                                                                                                                                                          |
| GAP              | Root node gap in percent, $GAP = (RUB - RLB)/RLB \cdot 100$ .                                                                                                                                                                                   |
| QUAL             | Quality of the root node lower bound in percent, $QUAL = 100 - (GUB - RLB)/RLB \cdot 100$ .                                                                                                                                                     |
| <b>Run</b>       |                                                                                                                                                                                                                                                 |
| #N               | Number of nodes in the branch and bound tree.                                                                                                                                                                                                   |
| CPU <sup>1</sup> | CPU time in seconds to find the first optimal solution. If no optimal solution is found within the given time limit, the CPU time for the current best solution is stated. If no feasible solution is found within the time limit, - is stated. |
| CPU*             | Total CPU time in seconds to close the gap and find the optimal solution. If no optimal solution is found within the time limit, - is stated.                                                                                                   |

Table 5.1: Key to model solution tables 5.2 - 5.9.

### 5.1.1 Results for the CPP model

To provide an overview of the solution procedure in the IVE environment, figures 5.1 and 5.2 illustrate the MIP search and the branch and bound tree for a  $C = 3$  and a  $C = 4$  instance run by the CPP model.

When solving the 3S.t instance, illustrated in figure 5.1, the first feasible solution is found after 0.2 seconds and the gap is closed after another second, resulting in a total CPU time of 1.2 seconds. The root node lower and upper bounds,  $RLB = 2.5$  and  $RUB = 13$ , and the gap between these are displayed at the branch and bound tree as well as the number of processed nodes.

In figure 5.2 (see page 115), representing the solution of the 4M2B.t instance, the MIP search illustrates the process of finding the three integer solutions, first with an

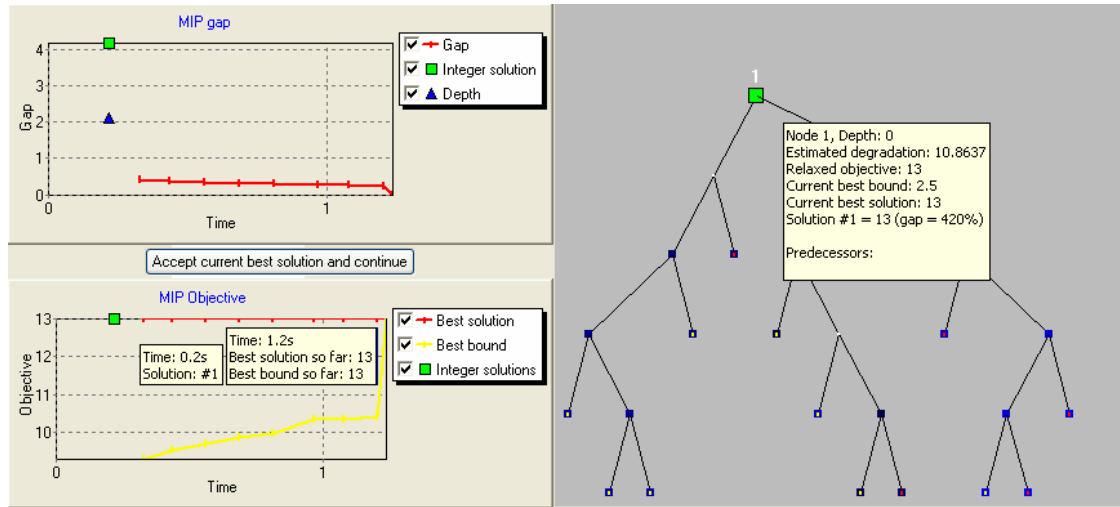


Figure 5.1: The Xpress MIP search and branch and bound tree for the 3S.t instance run with the CPP model. An optimal solution is found within 0.2 seconds and the gap is closed in a total of 1.2 seconds. The branch and bound tree shows the relaxed solution in the root node of 2.5 and the total number of nodes of 25.

objective of 18, the next 17, and the third and optimal solution with an objective value of 16. The optimal solution is found within 0.2 seconds but it takes another 2 seconds to close the gap. At the branch and bound tree with 381 nodes, the relaxed solution in the root node,  $RLB = 2.22$ , is quite poor and with the root upper bound,  $RUB = 18$ , the gap at the root node is 710 %.

Tables 5.2 and 5.3 state the results from the tight and scaled versions of the small-scale class problem instances, and tables 5.4 and 5.5 contain the results from the the medium-scale class. Each solution is documented in appendix D.

| Small<br>tight | Solution |    |      | Root |       |          |         | Run     |                  |         |
|----------------|----------|----|------|------|-------|----------|---------|---------|------------------|---------|
|                | OPT      | #M | GAP  | RLB  | RUB   | GAP      | QUAL    | #N      | CPU <sup>1</sup> | CPU*    |
| 3S.t           | 13       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -320.00 | 25      | 0.2              | 1.2     |
| 3C.t           | 9        | 3  | 0.00 | 2.50 | 15.00 | 500.00   | -160.00 | 1       | 0.1              | 0.1     |
| 3M1A.t         | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.5     |
| 3M1B.t         | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.4     |
| 3M2A.t         | 11       | 3  | 0.00 | 2.50 | 15.00 | 500.00   | -240.00 | 1       | 0.1              | 0.4     |
| 3M2B.t         | 11       | 3  | 0.00 | 2.50 | 15.00 | 500.00   | -240.00 | 1       | 0.1              | 0.4     |
| 3M3A.t         | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.4     |
| 3M3B.t         | 11       | 3  | 0.00 | 2.50 | 11.00 | 340.00   | -240.00 | 1       | 0.0              | 0.4     |
| 4S.t           | 20       | 6  | 0.00 | 2.22 | 21.00 | 845.00   | -700.00 | 1991    | 0.2              | 7.9     |
| 4C.t           | 12       | 4  | 0.00 | 2.22 | 12.00 | 440.01   | -340.01 | 1       | 0.1              | 0.2     |
| 4M1A.t         | 16       | 4  | 0.00 | 2.22 | 21.00 | 845.01   | -520.01 | 307     | 0.3              | 1.9     |
| 4M1B.t         | 17       | 5  | 0.00 | 2.22 | 21.00 | 845.01   | -565.01 | 449     | 0.3              | 2.8     |
| 4M2A.t         | 17       | 5  | 0.00 | 2.22 | 18.00 | 710.01   | -565.01 | 427     | 1.0              | 2.5     |
| 4M2B.t         | 16       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -520.01 | 381     | 0.2              | 2.2     |
| 4M3A.t         | 16       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -520.01 | 391     | 0.3              | 2.0     |
| 4M3B.t         | 14       | 4  | 0.00 | 2.22 | 15.00 | 575.01   | -430.01 | 9       | 0.3              | 0.5     |
| 4M4A.t         | 14       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -430.01 | 11      | 0.2              | 0.4     |
| 4M4B.t         | 14       | 4  | 0.00 | 2.22 | 15.00 | 575.01   | -430.01 | 21      | 0.2              | 0.5     |
| 5S.t           | 25       | 7  | 0.00 | 2.50 | 36.00 | 1,340.00 | -800.00 | 278,457 | 27.2             | 1,410.2 |
| 5C.t           | 15       | 5  | 0.00 | 2.50 | 15.00 | 500.00   | -400.00 | 1       | 0.1              | 0.5     |
| 5M1A.t         | 21       | 5  | 0.00 | 2.50 | 23.00 | 820.00   | -640.00 | 3,771   | 0.2              | 31.1    |
| 5M1B.t         | 21       | 5  | 0.00 | 2.50 | 23.00 | 820.00   | -640.00 | 5,093   | 0.2              | 41.4    |
| 5M2A.t         | 21       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -640.00 | 3,785   | 0.1              | 31.0    |
| 5M2B.t         | 19       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -560.00 | 883     | 0.4              | 8.7     |
| 5M3A.t         | 21       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -640.00 | 4,181   | 0.1              | 40.7    |
| 5M3B.t         | 19       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -560.00 | 1,169   | 0.2              | 13.1    |
| 5M4A.t         | 19       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -560.00 | 685     | 0.2              | 9.7     |
| 5M4B.t         | 19       | 5  | 0.00 | 2.50 | 19.00 | 660.00   | -560.00 | 1,373   | 0.1              | 15.9    |
| 5M5A.t         | 19       | 5  | 0.00 | 2.50 | 19.00 | 660.00   | -560.00 | 1,423   | 0.1              | 18.5    |
| 5M5B.t         | 19       | 5  | 0.00 | 2.50 | 19.00 | 660.00   | -560.00 | 973     | 0.1              | 13.4    |

Table 5.2: Results for the CPP model on small-scale class tight problems.



| Small<br>scaled | Solution |    |      | Root |       |          |         | Run     |                  |         |
|-----------------|----------|----|------|------|-------|----------|---------|---------|------------------|---------|
|                 | OPT      | #M | GAP  | RLB  | RUB   | GAP      | QUAL    | #N      | CPU <sup>1</sup> | CPU*    |
| 3S.s            | 14       | 4  | 0.00 | 2.50 | 18.00 | 620.00   | -360.00 | 19      | 0.2              | 0.8     |
| 3C.s            | 9        | 3  | 0.00 | 2.50 | 15.00 | 500.00   | -160.00 | 1       | 0.1              | 0.1     |
| 3M1A.s          | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.5     |
| 3M1B.s          | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.5     |
| 3M2A.s          | 11       | 3  | 0.00 | 2.50 | 11.00 | 340.00   | -240.00 | 1       | 0.0              | 0.5     |
| 3M2B.s          | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.4     |
| 3M3A.s          | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.6     |
| 3M3B.s          | 11       | 3  | 0.00 | 2.50 | 13.00 | 420.00   | -240.00 | 1       | 0.1              | 0.6     |
| 4S.s            | 20       | 6  | 0.00 | 2.22 | 21.00 | 845.00   | -700.00 | 2,167   | 5.5              | 8.5     |
| 4C.s            | 12       | 4  | 0.00 | 2.22 | 12.00 | 440.01   | -340.01 | 1       | 0.1              | 0.2     |
| 4M1A.s          | 17       | 5  | 0.00 | 2.22 | 21.00 | 845.01   | -565.01 | 599     | 3.0              | 3.2     |
| 4M1B.s          | 17       | 5  | 0.00 | 2.22 | 18.00 | 710.01   | -565.01 | 679     | 1.3              | 4.5     |
| 4M2A.s          | 17       | 5  | 0.00 | 2.22 | 18.00 | 710.01   | -565.01 | 389     | 0.3              | 2.6     |
| 4M2B.s          | 16       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -520.01 | 671     | 1.4              | 3.7     |
| 4M3A.s          | 16       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -520.01 | 393     | 0.3              | 2.2     |
| 4M3B.s          | 14       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -430.01 | 7       | 0.2              | 0.5     |
| 4M4A.s          | 14       | 4  | 0.00 | 2.22 | 18.00 | 710.01   | -430.01 | 87      | 0.8              | 0.9     |
| 4M4B.s          | 14       | 4  | 0.00 | 2.22 | 15.00 | 575.01   | -430.01 | 37      | 0.3              | 0.7     |
| 5S.s            | 25       | 7  | 0.00 | 3.33 | 40.00 | 1,100.01 | -550.01 | 278,109 | 0.5              | 1,464.7 |
| 5C.s            | 15       | 5  | 0.00 | 2.50 | 15.00 | 500.00   | -400.00 | 1       | 0.1              | 0.6     |
| 5M1A.s          | 21       | 5  | 0.00 | 2.50 | 23.00 | 820.00   | -640.00 | 2,975   | 6.2              | 22.5    |
| 5M1B.s          | 21       | 5  | 0.00 | 2.50 | 23.00 | 820.00   | -640.00 | 3,207   | 0.2              | 27.1    |
| 5M2A.s          | 21       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -640.00 | 6,845   | 0.1              | 52.4    |
| 5M2B.s          | 19       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -560.00 | 953     | 4.2              | 8.6     |
| 5M3A.s          | 21       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -640.00 | 12,527  | 0.1              | 80.9    |
| 5M3B.s          | 19       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -560.00 | 995     | 0.5              | 9.6     |
| 5M4A.s          | 19       | 5  | 0.00 | 2.50 | 21.00 | 740.00   | -560.00 | 1,487   | 0.2              | 12.6    |
| 5M4B.s          | 19       | 5  | 0.00 | 2.50 | 19.00 | 660.00   | -560.00 | 1,649   | 0.1              | 14.0    |
| 5M5A.s          | 19       | 5  | 0.00 | 2.50 | 19.00 | 660.00   | -560.00 | 857     | 0.1              | 9.4     |
| 5M5B.s          | 19       | 5  | 0.00 | 2.50 | 19.00 | 660.00   | -560.00 | 4,355   | 0.1              | 31.5    |

Table 5.3: Results for the CPP model on small-scale class scaled problems.

| Medium<br>tight | Solution   |    |       | Root  |          |          |         | Run     |                  |         |
|-----------------|------------|----|-------|-------|----------|----------|---------|---------|------------------|---------|
|                 | OPT        | #M | GAP   | RLB   | RUB      | GAP      | QUAL    | #N      | CPU <sup>1</sup> | CPU*    |
| 6S.t            | [27.00,31] | 9  | 14.81 | 17.50 | $\infty$ | -        | 22.86   | 422,559 | 65.1             | -       |
| 6C.t            | 18         | 6  | 0.00  | 3.00  | 18.00    | 500.00   | -400.00 | 1       | 0.1              | 0.5     |
| 6M1.t           | 26         | 6  | 0.00  | 17.50 | 28.00    | 60.00    | 51.43   | 26,717  | 0.7              | 149.0   |
| 6M2.t           | 26         | 6  | 0.00  | 3.00  | 26.00    | 766.67   | -666.67 | 54,525  | 0.1              | 268.8   |
| 6M3.t           | 22         | 6  | 0.00  | 3.00  | 22.00    | 633.33   | -533.33 | 1,493   | 0.1              | 11.8    |
| 6M4.t           | 24         | 6  | 0.00  | 3.00  | 24.00    | 700.00   | -600.00 | 10,385  | 0.1              | 60.5    |
| 6M5.t           | 24         | 6  | 0.00  | 15.33 | 40.00    | 160.87   | 43.48   | 3,519   | 3.9              | 20.7    |
| 6M6.t           | 22         | 6  | 0.00  | 3.00  | 22.00    | 633.33   | -533.33 | 3,649   | 0.1              | 19.9    |
| 8S.t            | [34.05,42] | 12 | 23.36 | 23.84 | $\infty$ | -        | 69.95   | 263,390 | 497.0            | -       |
| 8C.t            | 32         | 8  | 0.00  | 3.81  | 32.00    | 740.01   | -640.01 | 367     | 0.2              | 19.2    |
| 8M1.t           | [34.05,37] | 9  | 8.65  | 3.81  | 42.00    | 1,002.51 | -771.26 | 254,918 | 1,714.2          | -       |
| 8M2.t           | [34.03,38] | 10 | 11.67 | 3.81  | 40.00    | 950.01   | -797.51 | 234,312 | 747.3            | -       |
| 8M3.t           | 35         | 9  | 0.00  | 3.81  | 44.00    | 1,055.01 | -718.76 | 45,733  | 33.5             | 382.8   |
| 8M4.t           | 35         | 9  | 0.00  | 3.81  | 42.00    | 1,002.51 | -718.76 | 70,129  | 213.3            | 586.6   |
| 8M5.t           | 33         | 9  | 0.00  | 3.81  | 40.00    | 950.01   | -666.26 | 7,589   | 34.3             | 86.8    |
| 8M6.t           | 35         | 11 | 0.00  | 3.81  | 40.00    | 950.01   | -718.76 | 174,147 | 1,427.7          | 1,427.7 |
| 8M7.t           | 34         | 8  | 0.00  | 3.81  | 40.00    | 950.01   | -692.51 | 8,851   | 88.5             | 122.7   |
| 8M8.t           | 34         | 8  | 0.00  | 3.81  | 40.00    | 950.01   | -692.51 | 49,715  | 73.3             | 389.3   |
| 10S.t           | [40.31,52] | 14 | 29.00 | 29.93 | $\infty$ | -        | 26.29   | 131,309 | 1,217.9          | -       |
| 10C.t           | 40         | 10 | 0.00  | 4.76  | 40.00    | 740.00   | -640.00 | 1,235   | 0.2              | 53.2    |
| 10M1.t          | [41.15,47] | 11 | 14.21 | 4.76  | 50.00    | 950.00   | -787.00 | 192,272 | 882.5            | -       |
| 10M2.t          | [41.28,50] | 12 | 21.12 | 30.08 | $\infty$ | -        | 33.75   | 178,608 | 895.2            | -       |
| 10M3.t          | [41.09,45] | 11 | 9.53  | 4.76  | 54.00    | 1,034.00 | -745.00 | 163,630 | 1,207.6          | -       |
| 10M4.t          | [41.05,46] | 10 | 12.05 | 4.76  | 50.00    | 950.00   | -766.00 | 163,624 | 669.0            | -       |
| 10M5.t          | [41.03,44] | 12 | 7.24  | 4.76  | 46.00    | 866.00   | -724.00 | 137,786 | 314.7            | -       |
| 10M6.t          | 42         | 12 | 0.00  | 4.76  | 48.00    | 908.00   | -682.00 | 165,257 | 1,619.7          | 1,619.7 |
| 10M7.t          | [41.00,43] | 11 | 4.88  | 4.76  | 46.00    | 866.00   | -703.00 | 152,799 | 887.0            | -       |
| 10M8.t          | [41.01,43] | 11 | 9.72  | 29.99 | 56.00    | 86.74    | 56.61   | 141,028 | 1,675.5          | -       |
| 10M9.t          | [41.35,45] | 11 | 8.82  | 4.76  | 52.00    | 992.00   | -745.00 | 115,267 | 62.2             | -       |
| 10M10.t         | [41.00,45] | 11 | 7.14  | 4.76  | 48.00    | 908.00   | -745.00 | 131,593 | 1,307.4          | -       |

Table 5.4: Results for the CPP model on medium-scale class tight problems.

| Medium<br>scaled | Solution   |    |       | Root  |          |          |         | Run     |                  |       |
|------------------|------------|----|-------|-------|----------|----------|---------|---------|------------------|-------|
|                  | OPT        | #M | GAP   | RLB   | RUB      | GAP      | QUAL    | #N      | CPU <sup>1</sup> | CPU*  |
| 6S.s             | [25.08,31] | 9  | 23.62 | 4.00  | 47.00    | 1,075.00 | -575.00 | 479,419 | 195.6            | -     |
| 6C.s             | 18         | 6  | 0.00  | 3.00  | 18.00    | 500.00   | -400.00 | 1       | 0.1              | 0.5   |
| 6M1.s            | 27         | 7  | 0.00  | 3.00  | 46.00    | 1,433.33 | -700.00 | 97,993  | 16.1             | 504.2 |
| 6M2.s            | 26         | 6  | 0.00  | 3.00  | 26.00    | 766.67   | -666.67 | 19,765  | 0.1              | 103.1 |
| 6M3.s            | 24         | 6  | 0.00  | 3.00  | 24.00    | 700.00   | -600.00 | 5,637   | 0.1              | 36.6  |
| 6M4.s            | 24         | 6  | 0.00  | 3.00  | 24.00    | 700.00   | -600.00 | 11,163  | 0.1              | 61.0  |
| 6M5.s            | 24         | 6  | 0.00  | 4.00  | 52.00    | 1,200.00 | -400.00 | 18,053  | 5.7              | 95.3  |
| 6M6.s            | 22         | 6  | 0.00  | 3.00  | 22.00    | 633.33   | -533.33 | 1,515   | 0.1              | 12.3  |
| 8S.s             | [34.00,42] | 12 | 23.53 | 23.92 | 54.00    | 125.77   | 24.40   | 226,544 | 150.4            | -     |
| 8C.s             | 32         | 8  | 0.00  | 3.81  | 32.00    | 740.01   | -640.01 | 363     | 0.2              | 17.9  |
| 8M1.s            | [34.00,38] | 8  | 11.76 | 3.81  | 42.00    | 1,002.51 | -797.51 | 246,233 | 1.1              | -     |
| 8M2.s            | [34.00,38] | 10 | 11.76 | 3.81  | 40.00    | 950.01   | -797.51 | 242,766 | 1,785.4          | -     |
| 8M3.s            | 35         | 9  | 0.00  | 3.81  | 44.00    | 1,055.01 | -718.76 | 81,860  | 0.3              | 649.7 |
| 8M4.s            | 35         | 9  | 0.00  | 3.81  | 42.00    | 1,002.51 | -718.76 | 73,775  | 240.5            | 625.7 |
| 8M5.s            | 33         | 9  | 0.00  | 3.81  | 40.00    | 950.01   | -666.26 | 5,080   | 75.7             | 82.2  |
| 8M6.s            | [33.06,36] | 8  | 8.88  | 3.81  | 40.00    | 950.01   | -745.01 | 247,403 | 126.3            | -     |
| 8M7.s            | 34         | 8  | 0.00  | 3.81  | 40.00    | 950.01   | -692.51 | 40,653  | 28.9             | 324.2 |
| 8M8.s            | 34         | 8  | 0.00  | 3.81  | 36.00    | 845.01   | -692.51 | 34,223  | 2.8              | 295.3 |
| 10S.s            | [40.48,60] | 18 | 48.24 | 29.97 | 70.00    | 133.59   | -0.22   | 147,141 | 1,397.5          | -     |
| 10C.s            | 40         | 10 | 0.00  | 4.76  | 40.00    | 740.00   | -640.00 | 1,121   | 0.3              | 62.0  |
| 10M1.s           | [40.78,51] | 11 | 25.07 | 29.99 | $\infty$ | -        | 29.94   | 119,549 | 1,134.7          | -     |
| 10M2.s           | [40.72,54] | 12 | 32.61 | 29.93 | $\infty$ | -        | 19.59   | 147,060 | 88.3             | -     |
| 10M3.s           | [41.00,46] | 10 | 12.20 | 4.76  | 54.00    | 1,034.00 | -766.00 | 164,271 | 549.6            | -     |
| 10M4.s           | [40.81,46] | 10 | 12.71 | 4.76  | 52.00    | 992.00   | -766.00 | 182,239 | 523.7            | -     |
| 10M5.s           | [41.00,46] | 12 | 12.20 | 29.99 | $\infty$ | -        | 46.64   | 109,550 | 815.9            | -     |
| 10M6.s           | [41.00,45] | 11 | 9.76  | 4.76  | 48.00    | 908.00   | -745.00 | 182,701 | 351.2            | -     |
| 10M7.s           | [41.00,45] | 11 | 9.76  | 4.76  | 48.00    | 908.00   | -745.00 | 152,216 | 571.4            | -     |
| 10M8.s           | [40.38,49] | 11 | 21.35 | 30.00 | $\infty$ | -        | 36.67   | 124,198 | 184.9            | -     |
| 10M9.s           | [41.00,44] | 12 | 7.32  | 4.76  | 52.00    | 992.00   | -724.00 | 186,641 | 36.7             | -     |
| 10M10.s          | [41.00,44] | 12 | 7.32  | 29.98 | $\infty$ | -        | 53.22   | 193,173 | 264.6            | -     |

Table 5.5: Results for the CPP model on medium-scale class scaled problems.

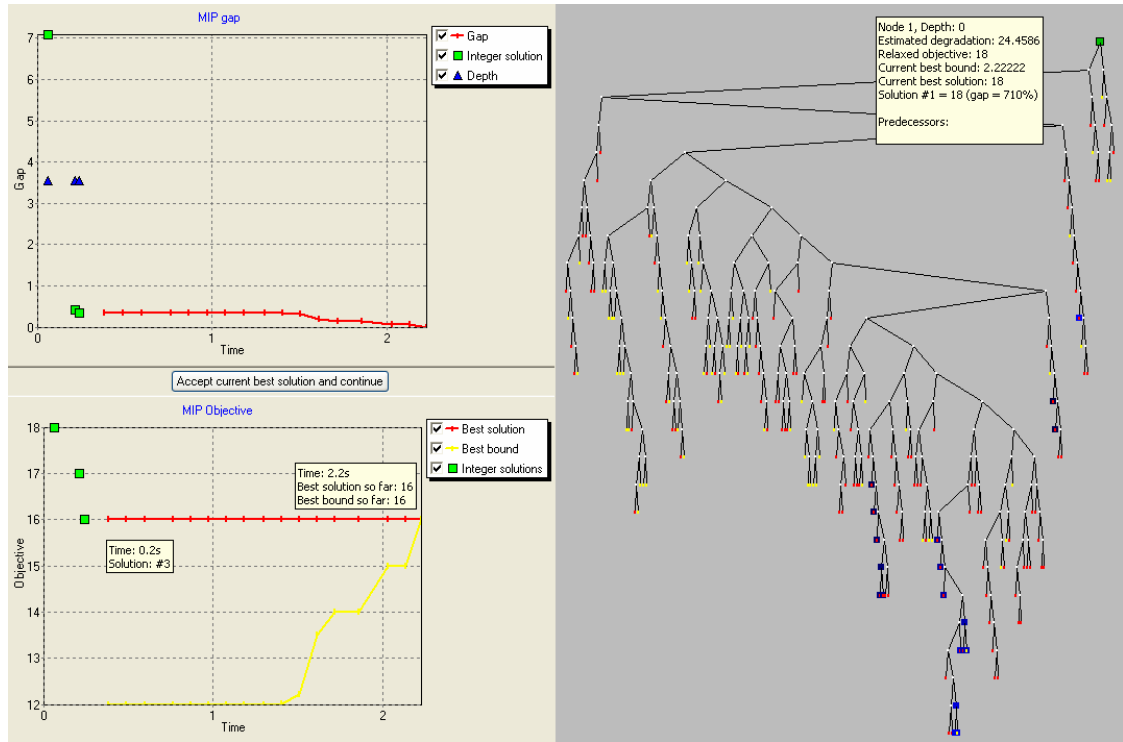


Figure 5.2: The Xpress MIP search and branch and bound tree for the 4M2B.t instance run with the CPP model. An optimal solution is found within 0.2 seconds and the gap is closed in a total of 2.2 seconds. The 381 nodes branch and bound tree shows the relaxed solution in the root node of 2.22 and the upper bound of 18, resulting in a gap of 710 %.

## Discussion of the CPP results

All small-scale problem instances are solved to optimality within the time limit. The stair instances are significantly harder to solve, requiring longer run times than the other types and, in general, the cone instances are easiest to solve.

In the 3 containers subclass, all tight as well as scaled problems except the stair instances are solved in the root node, all with a total CPU time under 1.0, the 3S.t instance being an exception with CPU\* = 1.2 seconds. In the 4 containers subclass, solution times vary between under 1.0 second and under 10.0 seconds, again the stair instances being the most time-consuming. Only the cone instances - both with CPU times of 0.2 seconds - are solved to optimality in the root node. Run times for the stair instances increase dramatically in the 5 container subclass. Both the tight and scaled versions of the stair instance solve to optimality in over 20 minutes whereas the other problem types solve in about 1.0 minute, again the cone instances distinguishing by solution times under 1.0 second and no need for branching.

There is no significant difference in run times between the tight and scaled versions.

Common to all small-scale instances, tight as well as scaled, is that the first optimal solution is found within a fraction of a second in most cases, exceptions being two stair instances and a few of the 4 and 5 containers scaled instances. The quality of the root node bound is, in general, poor and decreasing by growing problem size which also shows in the quality, varying between -160 and -800.

Contrary to the small-scale class results, not all medium-scale instances are solved to optimality within the time limit. Again, the stair types require longer solution times and none of these six instances results in a closed gap whereas all cone problems solve to optimality quite fast. In some cases, no feasible solution is found in the root node, i.e.  $RUB = \infty$ .

In the 6 containers subclass, all 16 problems except for the stair instances solve to optimality and the first optimal solution is found within a fraction of a second with a few exceptions. In the 8 containers subclass, three of the tight and four of the scaled instances do not solve to optimality within the time limit. With a few exceptions, run times are counted in minutes rather than seconds, again with the cone instances solved fastest, and none of the problems are solved to optimality in the root node. Only three of the 10 containers subclass problems are solved to optimality within the time limit, two of them being the cone instances. Also, the current best solution when optimality is not reached is found after a considerable run time, and only in the cone instances, the first optimal solution is found in less than 1.0 second.

Equivalent to the small-scale results, there is no appreciable difference in run times between the tight and scaled versions of the medium-scale class problems, the reason presumably being that the CPP model is not sensitive to extension of the time dimension as the size in variables and constraints does not increase by changing the time parameters. Contrary to the small-scale class, RLB results in a positive quality in 13 cases of the medium-scale instances. However, the root node solution quality is still poor in many cases, approaching the worst case quality of -800, observed in the small-scale test phase.

In general, the CPP model performs well on the small-scale class problems but when increasing the number of containers, the performance deteriorates significantly. This confirms the expected conclusion that solving the CPP model using standard software is not an efficient approach to the problem. Results for the CPP model is compared to those for the CPPT model and the heuristic in section 5.3 on page 203.

### 5.1.2 Results for the CPPT model

As well as the CPP model, the CPPT model is validated by solving the 30 tight and 30 scaled small-scale problem instances and its performance is evaluated by the 60 medium-scale class instances. The results are displayed tables 5.6 - 5.9 and keys to the tables can be found in table 5.1 on page 109. Each solution is documented in appendix E.

As an example of the MIP search in the IVE environment, figure 5.3 illustrates the process of closing the gap in the 6S.t instance, solved by the CPPT model in 1,068.7 seconds. Even though the optimal solution is found in 20.9 seconds, the lower bound improves so slowly that it takes almost 20 minutes to close the gap. The reason for this may be the symmetry of the problem, resulting in a large number of permuted

solutions. The branch and bound tree, consisting of 18,157 nodes, is not depicted in the figure.

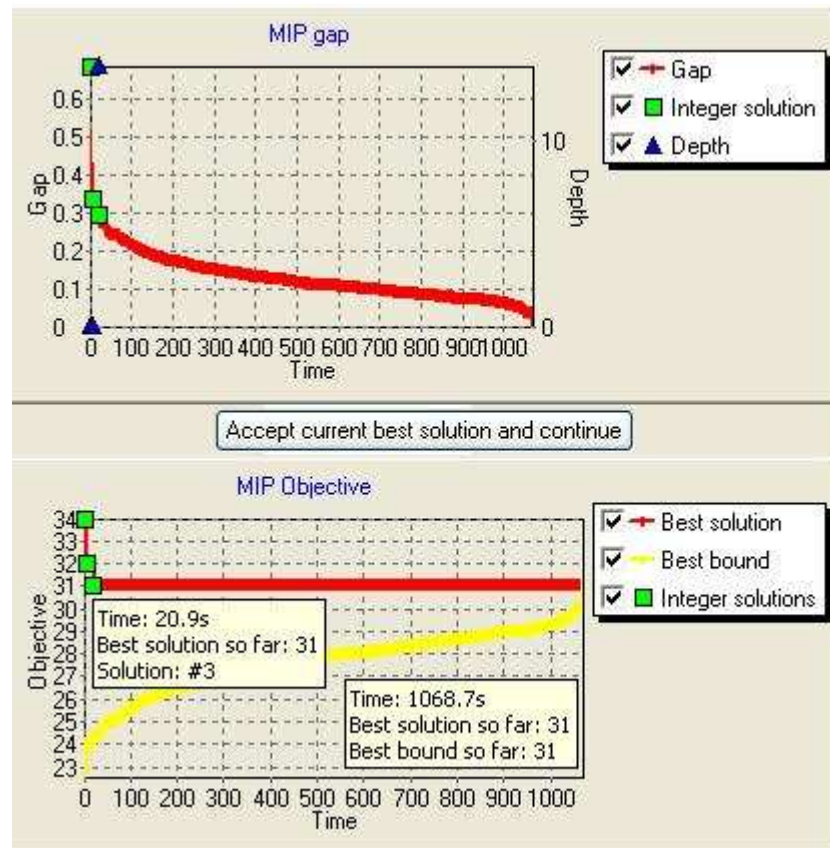


Figure 5.3: The Xpress MIP for the 6S.t instance run with the CPPT model. An optimal solution is found within 20.9 seconds but more than 1,000 seconds are required to close the gap which may be due to problem symmetry.

| Small<br>tight | Solution |    |      | Root  |       |       |        | Run   |                  |      |
|----------------|----------|----|------|-------|-------|-------|--------|-------|------------------|------|
|                | OPT      | #M | GAP  | RLB   | RUB   | GAP   | QUAL   | #N    | CPU <sup>1</sup> | CPU* |
| 3S.t           | 14       | 4  | 0.00 | 14.00 | 14.00 | 0.00  | 100.00 | 1     | 0.1              | 0.1  |
| 3C.t           | 11       | 3  | 0.00 | 11.00 | 11.00 | 0.00  | 100.00 | 1     | 0.0              | 0.0  |
| 3M1A.t         | 15       | 5  | 0.00 | 13.28 | 15.00 | 12.99 | 87.01  | 1     | 0.0              | 0.1  |
| 3M1B.t         | 11       | 3  | 0.00 | 10.73 | 12.00 | 11.80 | 97.52  | 1     | 0.0              | 0.1  |
| 3M2A.t         | 11       | 3  | 0.00 | 11.00 | 11.00 | 0.00  | 100.00 | 1     | 0.0              | 0.0  |
| 3M2B.t         | 11       | 3  | 0.00 | 11.00 | 11.00 | 0.00  | 100.00 | 1     | 0.0              | 0.0  |
| 3M3A.t         | 12       | 4  | 0.00 | 12.00 | 12.00 | 0.00  | 100.00 | 1     | 0.0              | 0.0  |
| 3M3B.t         | 11       | 3  | 0.00 | 10.77 | 11.00 | 2.10  | 97.90  | 1     | 0.0              | 0.0  |
| 4S.t           | 20       | 6  | 0.00 | 14.61 | 20.00 | 36.86 | 63.14  | 23    | 0.1              | 1.2  |
| 4C.t           | 14       | 4  | 0.00 | 13.79 | 15.00 | 8.81  | 98.45  | 1     | 0.1              | 0.2  |
| 4M1A.t         | 16       | 4  | 0.00 | 14.61 | 17.00 | 16.38 | 90.46  | 1     | 0.3              | 0.3  |
| 4M1B.t         | 17       | 5  | 0.00 | 14.79 | 18.00 | 21.74 | 85.02  | 1     | 0.1              | 0.2  |
| 4M2A.t         | 17       | 5  | 0.00 | 14.83 | 17.00 | 14.67 | 85.33  | 1     | 0.1              | 0.1  |
| 4M2B.t         | 16       | 4  | 0.00 | 14.83 | 18.00 | 21.42 | 92.07  | 1     | 0.1              | 0.2  |
| 4M3A.t         | 16       | 4  | 0.00 | 14.90 | 17.00 | 14.09 | 92.62  | 1     | 0.1              | 0.1  |
| 4M3B.t         | 14       | 4  | 0.00 | 13.73 | 15.00 | 9.22  | 98.06  | 1     | 0.1              | 0.1  |
| 4M4A.t         | 14       | 4  | 0.00 | 13.85 | 15.00 | 8.33  | 98.90  | 1     | 0.1              | 0.1  |
| 4M4B.t         | 14       | 4  | 0.00 | 13.85 | 15.00 | 8.33  | 98.90  | 1     | 0.1              | 0.2  |
| 5S.t           | 25       | 7  | 0.00 | 17.11 | 25.00 | 46.09 | 53.91  | 1,211 | 0.7              | 66.6 |
| 5C.t           | 19       | 5  | 0.00 | 16.20 | 24.00 | 48.18 | 82.69  | 1     | 1.0              | 1.4  |
| 5M1A.t         | 21       | 5  | 0.00 | 17.71 | 23.00 | 29.87 | 81.42  | 27    | 4.7              | 4.7  |
| 5M1B.t         | 21       | 5  | 0.00 | 17.48 | 22.00 | 25.88 | 79.84  | 1     | 1.9              | 1.9  |
| 5M2A.t         | 21       | 5  | 0.00 | 17.41 | 24.00 | 37.88 | 79.35  | 1     | 2.5              | 2.5  |
| 5M2B.t         | 19       | 5  | 0.00 | 17.29 | 22.00 | 27.21 | 90.14  | 1     | 0.6              | 0.7  |
| 5M3A.t         | 21       | 5  | 0.00 | 17.60 | 24.00 | 36.39 | 80.66  | 1     | 0.7              | 1.4  |
| 5M3B.t         | 19       | 5  | 0.00 | 17.30 | 21.00 | 21.42 | 90.14  | 1     | 0.6              | 0.8  |
| 5M4A.t         | 19       | 5  | 0.00 | 17.27 | 22.00 | 27.37 | 90.00  | 1     | 0.6              | 0.7  |
| 5M4B.t         | 19       | 5  | 0.00 | 17.33 | 22.00 | 26.97 | 90.34  | 1     | 0.6              | 0.7  |
| 5M5A.t         | 19       | 5  | 0.00 | 16.64 | 19.00 | 14.15 | 85.85  | 1     | 0.5              | 1.2  |
| 5M5B.t         | 19       | 5  | 0.00 | 16.78 | 23.00 | 37.08 | 86.76  | 1     | 0.6              | 0.9  |

Table 5.6: Results for the CPPT model on small-scale class tight problems.

| Small<br>scaled | Solution   |    |       | Root  |       |        |        | Run |                  |         |
|-----------------|------------|----|-------|-------|-------|--------|--------|-----|------------------|---------|
|                 | OPT        | #M | GAP   | RLB   | RUB   | GAP    | QUAL   | #N  | CPU <sup>1</sup> | CPU*    |
| 3S.s            | 14         | 4  | 0.00  | 13.21 | 15.00 | 13.51  | 94.05  | 1   | 0.3              | 0.5     |
| 3C.s            | 11         | 3  | 0.00  | 11.00 | 11.00 | 0.00   | 100.00 | 1   | 0.3              | 0.3     |
| 3M1A.s          | 15         | 5  | 0.00  | 13.18 | 18.00 | 36.57  | 86.19  | 1   | 0.3              | 0.5     |
| 3M1B.s          | 11         | 3  | 0.00  | 10.95 | 12.00 | 9.56   | 99.57  | 1   | 0.6              | 0.8     |
| 3M2A.s          | 11         | 3  | 0.00  | 10.99 | 12.00 | 9.22   | 99.88  | 1   | 0.2              | 0.3     |
| 3M2B.s          | 11         | 3  | 0.00  | 10.98 | 12.00 | 9.26   | 99.84  | 1   | 0.3              | 0.3     |
| 3M3A.s          | 12         | 4  | 0.00  | 12.00 | 12.00 | 0.00   | 100.00 | 1   | 0.3              | 0.3     |
| 3M3B.s          | 11         | 3  | 0.00  | 10.96 | 12.00 | 9.50   | 99.62  | 1   | 0.4              | 0.6     |
| 4S.s            | 20         | 6  | 0.00  | 14.65 | 21.00 | 43.35  | 63.48  | 153 | 8.6              | 294.8   |
| 4C.s            | 14         | 4  | 0.00  | 13.97 | 15.00 | 7.38   | 99.78  | 1   | 3.6              | 4.0     |
| 4M1A.s          | 17         | 5  | 0.00  | 14.65 | 18.00 | 22.88  | 83.95  | 3   | 7.5              | 23.3    |
| 4M1B.s          | 17         | 5  | 0.00  | 14.65 | 21.00 | 43.39  | 83.93  | 5   | 4.0              | 14.3    |
| 4M2A.s          | 17         | 5  | 0.00  | 14.00 | 17.00 | 21.43  | 78.57  | 7   | 2.1              | 24.5    |
| 4M2B.s          | 16         | 4  | 0.00  | 14.00 | 16.00 | 14.29  | 85.71  | 1   | 1.8              | 16.9    |
| 4M3A.s          | 16         | 4  | 0.00  | 14.00 | 16.00 | 14.29  | 85.71  | 3   | 2.0              | 17.4    |
| 4M3B.s          | 14         | 4  | 0.00  | 13.97 | 18.00 | 28.88  | 99.76  | 1   | 2.2              | 2.4     |
| 4M4A.s          | 14         | 4  | 0.00  | 13.97 | 15.00 | 7.40   | 99.76  | 1   | 2.8              | 3.1     |
| 4M4B.s          | 14         | 4  | 0.00  | 13.97 | 15.00 | 7.37   | 99.79  | 1   | 2.9              | 3.3     |
| 5S.s            | [19.00,26] | 8  | 36.84 | 15.96 | 39.00 | 144.41 | 37.06  | 34  | 1,373.2          | -       |
| 5C.s            | 19         | 5  | 0.00  | 15.29 | 33.00 | 115.81 | 75.74  | 11  | 977.8            | 977.8   |
| 5M1A.s          | [19.20,22] | 6  | 14.59 | 15.87 | 36.00 | 126.88 | 61.35  | 67  | 619.4            | -       |
| 5M1B.s          | [19.67,21] | 5  | 6.78  | 15.86 | 35.00 | 120.65 | 67.61  | 56  | 440.1            | -       |
| 5M2A.s          | 21         | 5  | 0.00  | 15.83 | 30.00 | 89.48  | 67.36  | 99  | 1,248.6          | 1,852.4 |
| 5M2B.s          | 19         | 5  | 0.00  | 15.64 | 30.00 | 91.78  | 78.54  | 4   | 246.7            | 246.7   |
| 5M3A.s          | [20.00,21] | 5  | 5.00  | 15.96 | 33.00 | 106.81 | 68.39  | 68  | 497.9            | -       |
| 5M3B.s          | 19         | 5  | 0.00  | 15.85 | 33.00 | 108.15 | 80.16  | 29  | 635.3            | 635.3   |
| 5M4A.s          | 19         | 5  | 0.00  | 15.95 | 36.00 | 125.71 | 80.87  | 3   | 547.3            | 547.3   |
| 5M4B.s          | 19         | 5  | 0.00  | 15.83 | 19.00 | 20.02  | 79.98  | 1   | 59.1             | 388.0   |
| 5M5A.s          | 19         | 5  | 0.00  | 15.60 | 30.00 | 92.30  | 78.21  | 3   | 518.5            | 518.5   |
| 5M5B.s          | 19         | 5  | 0.00  | 15.85 | 33.00 | 108.23 | 80.11  | 3   | 531.2            | 531.2   |

Table 5.7: Results for the CPPT model on small-scale class scaled problems.



| Medium<br>tight | Solution   |    |        | Root  |          |        |        | Run    |                  |         |
|-----------------|------------|----|--------|-------|----------|--------|--------|--------|------------------|---------|
|                 | OPT        | #M | GAP    | RLB   | RUB      | GAP    | QUAL   | #N     | CPU <sup>1</sup> | CPU*    |
| 6S.t            | 31         | 9  | 0.00   | 20.16 | 34.00    | 68.63  | 46.25  | 18,157 | 20.9             | 1,068.7 |
| 6C.t            | 24         | 6  | 0.00   | 19.31 | 26.00    | 34.62  | 75.74  | 1      | 2.1              | 2.2     |
| 6M1.t           | 27         | 7  | 0.00   | 20.49 | 30.00    | 46.38  | 68.25  | 59     | 8.5              | 9.9     |
| 6M2.t           | 26         | 6  | 0.00   | 20.29 | 28.00    | 38.02  | 71.84  | 21     | 1.5              | 9.4     |
| 6M3.t           | 22         | 6  | 0.00   | 20.06 | 22.00    | 9.65   | 90.35  | 1      | 0.4              | 0.6     |
| 6M4.t           | 24         | 6  | 0.00   | 20.52 | 29.00    | 41.35  | 83.02  | 1      | 1.2              | 1.2     |
| 6M5.t           | 24         | 6  | 0.00   | 20.16 | 26.00    | 28.97  | 80.95  | 1      | 1.9              | 1.9     |
| 6M6.t           | 24         | 6  | 0.00   | 23.00 | 24.00    | 4.35   | 95.65  | 25     | 7.7              | 7.7     |
| 8S.t            | [36.00,51] | 17 | 41.67  | 35.15 | 51.00    | 45.10  | 54.90  | 384    | 197.9            | -       |
| 8C.t            | [36.03,38] | 10 | 5.47   | 26.45 | 44.00    | 66.35  | 56.33  | 852    | 93.3             | -       |
| 8M1.t           | [36.00,40] | 10 | 11.11  | 36.00 | $\infty$ | -      | 88.89  | 685    | 739.4            | -       |
| 8M2.t           | [36.07,39] | 11 | 8.12   | 36.00 | $\infty$ | -      | 91.67  | 590    | 1,001.7          | -       |
| 8M3.t           | [36.06,42] | 12 | 16.46  | 25.65 | 43.00    | 67.65  | 36.25  | 1,019  | 241.4            | -       |
| 8M4.t           | 37         | 11 | 0.00   | 26.38 | 42.00    | 59.21  | 59.74  | 209    | 122.7            | 163.7   |
| 8M5.t           | 37         | 11 | 0.00   | 34.12 | 44.00    | 28.97  | 91.55  | 499    | 328.3            | 357.4   |
| 8M6.t           | 37         | 9  | 0.00   | 35.50 | $\infty$ | -      | 95.77  | 223    | 190.6            | 201.1   |
| 8M7.t           | 38         | 10 | 0.00   | 32.42 | 43.00    | 32.63  | 82.79  | 561    | 345.5            | 681.7   |
| 8M8.t           | 38         | 10 | 0.00   | 26.14 | 38.00    | 45.37  | 54.63  | 1,251  | 5.8              | 1,432.3 |
| 10S.t           | [40.54,58] | 18 | 43.08  | 30.62 | 70.00    | 128.58 | 10.61  | 322    | 657.8            | -       |
| 10C.t           | [31.20,72] | 22 | 130.76 | 31.20 | 84.00    | 169.22 | -30.76 | 1      | 227.0            | -       |
| 10M1.t          | [40.04,57] | 17 | 42.36  | 31.37 | 83.00    | 164.62 | 18.27  | 143    | 1,592.1          | -       |
| 10M2.t          | [40.01,53] | 13 | 32.47  | 31.50 | 61.00    | 93.64  | 31.76  | 189    | 1,003.2          | -       |
| 10M3.t          | [40.09,47] | 11 | 17.23  | 31.42 | 64.00    | 103.72 | 50.39  | 147    | 1,085.0          | -       |
| 10M4.t          | [39.12,65] | 19 | 66.16  | 31.24 | 65.00    | 108.08 | -8.08  | 1      | 86.7             | -       |
| 10M5.t          | [40.21,51] | 15 | 26.82  | 31.26 | 58.00    | 85.56  | 36.83  | 101    | 900.4            | -       |
| 10M6.t          | [40.01,45] | 11 | 12.47  | 31.90 | 62.00    | 94.35  | 58.94  | 114    | 1,392.5          | -       |
| 10M7.t          | [40.16,43] | 11 | 7.08   | 31.92 | 69.00    | 116.16 | 65.29  | 79     | 1,702.5          | -       |
| 10M8.t          | [40.02,58] | 16 | 44.94  | 31.05 | 72.00    | 131.91 | 13.19  | 4      | 1,749.5          | -       |
| 10M9.t          | [30.97,65] | 19 | 109.88 | 30.97 | 88.00    | 184.15 | -9.88  | 1      | 169.3            | -       |
| 10M10.t         | [31.03,72] | 22 | 132.00 | 31.03 | 72.00    | 132.00 | -32.00 | 1      | 300.6            | -       |

Table 5.8: Results for the CPPT model on medium-scale class tight problems.

| Medium<br>scaled | Solution   |    |        | Root  |       |        |        | Run |                  |         |
|------------------|------------|----|--------|-------|-------|--------|--------|-----|------------------|---------|
|                  | OPT        | #M | GAP    | RLB   | RUB   | GAP    | QUAL   | #N  | CPU <sup>1</sup> | CPU*    |
| 6S.s             | [22.11,45] | 15 | 103.54 | 19.16 | 45.00 | 134.89 | -34.89 | 1   | 617.8            | -       |
| 6C.s             | [21.76,32] | 10 | 47.09  | 18.35 | 45.00 | 145.26 | 25.59  | 1   | 254.4            | -       |
| 6M1.s            | [24.00,42] | 14 | 75.00  | 24.00 | 44.00 | 83.33  | 25.00  | 3   | 1,659.5          | -       |
| 6M2.s            | [24.00,34] | 10 | 41.67  | 18.91 | 39.00 | 106.24 | 20.20  | 4   | 1,669.9          | -       |
| 6M3.s            | [24.00,38] | 12 | 58.33  | 24.00 | 38.00 | 58.33  | 41.67  | 26  | 978.6            | -       |
| 6M4.s            | [24.00,42] | 14 | 75.00  | 18.84 | 42.00 | 122.99 | -22.99 | 1   | 141.5            | -       |
| 6M5.s            | 24         | 6  | 0.00   | 18.97 | 33.00 | 73.96  | 73.49  | 3   | 1,235.4          | 1,235.6 |
| 6M6.s            | 24         | 6  | 0.00   | 18.62 | 38.00 | 104.12 | 71.08  | 6   | 1,743.9          | 1,744.1 |
| 8S.s             | -          | -  | -      | -     | -     | -      | -      | -   | -                | (c)     |
| 8C.s             | -          | -  | -      | -     | -     | -      | -      | -   | -                | (e)     |
| 8M1.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (d)     |
| 8M2.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (d)     |
| 8M3.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (d)     |
| 8M4.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (e)     |
| 8M5.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (e)     |
| 8M6.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (e)     |
| 8M7.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (e)     |
| 8M8.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (e)     |
| 10S.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |
| 10C.s            | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |
| 10M1.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (b)     |
| 10M2.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (b)     |
| 10M3.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (b)     |
| 10M4.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |
| 10M5.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (b)     |
| 10M6.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |
| 10M7.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (b)     |
| 10M8.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |
| 10M9.s           | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |
| 10M10.s          | -          | -  | -      | -     | -     | -      | -      | -   | -                | (a)     |

Table 5.9: Results for the CPPT model on medium-scale class scaled problems. (a) Out of memory while generating the matrix. (b) Out of memory while loading the problem. (c) Out of memory during pre-solve. (d) Out of memory during optimization. (e) Exceeding time limit while solving LP relaxation.

## Discussion of the CPPT results

Contrary to solutions for the CPP model, not all small-scale problem instances are solved to optimality by the CPPT model within the 1,800 seconds time limit. The 3 and 4 containers subclass problems as well as the tight versions of the 5 container subclass instances all solve far faster than the 30 minutes available but in four cases of the 5 containers scaled problems, the CPPT model run does not succeed in closing the gap. Furthermore, when solving the scaled instances with 5 containers, the time to find the first optimal solution or the current best solution in case of  $GAP > 0.0$ , stated in the CPU<sup>1</sup> column, increases to several hundred seconds in all cases except one instance.

These observations strongly indicate the CPPT model's sensitivity to expansion of the time dimension, leading to a heavy increase of the number of variables and constraints due to the time discretization in which the CPPT model differs from the CPP model. This conclusion is furthermore supported by the evident tendency of increased solution times for larger instances with a significantly larger time span. The relation is especially clear when comparing solution times for the tight and the scaled problem instances where the number of time slots is increased by a factor of about three. Such a distinct increase in run times when scaling the problem instances does not occur in the CPP model tests.

However, all small-scale tight problems except for a single instance are solved faster by the CPPT model than by the CPP model. All 3 containers instances tight as well as scaled are solved to optimality in the root within a fraction of a second, several of them in less than 0.1 seconds. In the 4 containers subclass all tight instances except the stair type are solved to optimality in the root node in around 0.2 seconds but when solving the scaled instances, run times increase to around 20 seconds in many cases and the stair instance solves in almost 300 seconds which is quite remarkable compared to the tight version with CPU\* = 1.2 seconds. In the 5 containers subclass all tight instances but two are solved to optimality in the root node which is the case for only one scaled instance. Furthermore, run times increase significantly for the scaled problems and four instances are not solved to optimality within the 1,800 seconds time limit.

In general, the CPPT model performs better than the CPP model when solving the small-scale tight problems but worse when solving scaled instances. As for the CPP model, the stair instances are harder to solve, requiring much longer run times, if even reaching optimum within the time limit. This difference between the problem types stands out more clearly the larger the problem instances. However, the quality of the root node lower bound is considerably better than for the CPP model, resulting in QUAL close to 100 % for many small-scale instances and only under 50 % in a single case.

In the medium-scale class the CPPT model shows a poor performance when increasing the size in number of containers as well as the time dimension. Only the tight 6 containers instances are solved to optimality within the time limit, run times increasing as would be expected from the progress in solution times for the tight small-scale instances. Furthermore, in the 8 and 10 containers subclasses no feasible solution is found to any of the scaled instances, and in many cases memory runs out before even starting the optimization, i.e. during the matrix generation, while loading the problem or when performing the pre-solve procedures.

In the 6 containers subclass, all tight instances are solved to optimality within the time limit, again the stair type requiring the longest run time. Only two of the scaled instances solve to optimality within 30 minutes and the gaps are very large, varying between 41.67 % and 103.54 %, the latter in the stair instance. Only half of the tight instances in the 8 containers subclass solve to optimality within the time limit, gaps varying between 5.47 % and 41.67 %, again the poorest results achieved for the stair instance. When solving the scaled versions of the 8 containers subclass problems, no feasible solutions are found, either due to heavy time consumption when solving the LP relaxation or for lack of memory while pre-solving or starting the actual optimization. In the 10 containers subclass, none of the tight problem instances are solved to optimality within the time limit and when solving the scaled instances, memory runs out even before the pre-solve phase.

Contrary to the small-scale results, the number of instances where optimum is not reached within the time limit or where memory runs out before finding a feasible solution exceeds the number of instances solved to optimality within 30 minutes. The quality of the relaxed root node solution decreases compared to the small-scale test phase but QUAL still surpasses the CPP model's performance in the root node. In many cases, however, solving the root node consumes several minutes, leaving little time to investigate other solutions.

In general, the CPPT model performs well in terms of run times and root node solution quality when solving very small problem instances. However, when increasing the problem size - especially the number of time slots - the model performs poorly compared to the CPP model as well as from a general point of view. Run times increase considerably for scaled versions compared to tight instances and when solving problems with just 8 and 10 containers, the model does not succeed in finding feasible solutions to instances with an extended time horizon. Running out of memory is, naturally, connected with computer power and performing the CPPT runs on a faster machine would, undoubtedly produce better results. However, the models - as well as the heuristic algorithm - are compared on the same hardware terms and it must, therefore, be concluded that the CPPT model is not suited for solving larger problem instances as it performs even worse than the CPP model when extending the time dimension of the instances. Although the LP relaxation of the CPPT model is quite good, resulting in high-quality lower bounds compared to the CPP model, which indicates a potential for finding optimum relatively fast, the convergence is very slow due to the extensive search in the branch and bound tree, exploding in size as the number of time slots increases.

These observations lead to the conclusion that solving the CPPT model using standard software is not suitable for handling the problem investigated in this research study but, rather, alternative approaches to finding high-quality solutions in short run times should be investigated. Results for the CPPT model are compared to the CPP model and heuristic results in section 5.3 on page 203.

### 5.1.3 Results for the models with extended run time limit

The main solution process is based on the assumption that little time is available to plan the container positioning, hence the run time limit of 1,800 seconds. However, if information about the ships' arrival and departure times as well as exact gate times for trucks and trains could become available earlier, it might be possible to obtain better solutions for the models. In order to investigate the potential gain of having the required information well in advance of containers arriving or departing, selected problem instances from the 10 containers subclass are run with an extended time limit of 28,800 seconds, corresponding to an eight hours run time, allowing planning e.g. over night or during a working day.

The possible gain in solution quality is investigated for both models. Results for the CPP model indicate that re-running scaled instances from the 10 containers subclass may provide interesting results (see table 5.5 on page 114). For lack of memory when running the CPPT model on the 10 containers scaled instances (see table 5.9 on page 121), tight versions from the subclass are tested in this investigation. The choice falls on the stair instances, proving to be the most time-consuming problem types, and the first of the mix instances, omitting the cone types as they solve to optimality within the original time limit when running the CPP model. Consequently, the 10S.s and 10M1.s are solved by the CPP model and the 10S.t and 10M1.t are solved by the CPPT model with an upper bound of 28,800 seconds on the total CPU time. The solutions are documented in appendix F.

Table 5.10 presents the results for the re-runs with the extended time limit. Each row corresponds to a single run, rows denoted by the instance name repeating the original results with the 1,800 seconds time limit and rows starting with -28,800 stating the results for the re-run with the 28,800 seconds time limit.

| CPP &<br>CPPT | Solution   |    |       | Gain              |                   |                   | Run       |                  |      |
|---------------|------------|----|-------|-------------------|-------------------|-------------------|-----------|------------------|------|
|               | OPT        | #M | GAP   | GLB <sup>im</sup> | GUB <sup>im</sup> | GAP <sup>im</sup> | #N        | CPU <sup>1</sup> | CPU* |
| 10S.s         | [40.48,60] | 18 | 48.24 | -                 | -                 | -                 | 147,141   | 1,397.5          | -    |
| - 28,800      | [42.04,54] | 16 | 28.44 | 3.88              | 10.00             | 38.77             | 3,100,672 | 18,668.0         | -    |
| 10M1.s        | [40.78,51] | 11 | 25.07 | -                 | -                 | -                 | 119,549   | 1,134.7          | -    |
| - 28,800      | [43.00,48] | 12 | 11.63 | 5.45              | 5.88              | 51.08             | 2,465,760 | 5,951.9          | -    |
| 10S.t         | [40.54,58] | 18 | 43.08 | -                 | -                 | -                 | 322       | 657.8            | -    |
| - 28,800      | [42.16,54] | 18 | 28.08 | 4.01              | 6.90              | 32.22             | 7,576     | 12,061.6         | -    |
| 10M1.t        | [40.04,57] | 17 | 42.36 | -                 | -                 | -                 | 143       | 1,592.1          | -    |
| - 28,800      | [41.27,50] | 14 | 21.14 | 3.09              | 12.28             | 48.56             | 4,348     | 25,111.6         | -    |

Table 5.10: Results for four problem instances from the 10 containers subclass, run with a 28,800 seconds time limit, 10S.s and 10M1.s solved by the CPP model and 10S.t and 10M1.t solved by the CPPT model.

The three **Solution** columns and the three **Run** columns represent [GLB,GUB], the number of moves, the optimality gap, the number of branch and bound nodes, CPU time for the current best solution, and the total CPU time. Each of these keys are explained in

the section introduction and in table 5.1 on page 109. Furthermore, three **Gain** columns state the advantage of extending the time limit, concerning improvements in solution quality. With GLB, GUB, and GAP denoting the original global lower bound, global upper bound, and gap and  $GLB^{28,800}$ ,  $GUB^{28,800}$ , and  $GAP^{28,800}$  denoting the corresponding results for the extended time limit re-runs, the cell values in the **Gain** columns are computed as follows.  $GLB^{im} = (GLB^{28,800} - GLB)/GLB \cdot 100$ ,  $GUB^{im} = (GUB - GUB^{28,800})/GUB \cdot 100$ , and  $GAP^{im} = ((GUB - GLB) - (GUB^{28,800} - GLB^{28,800})) / ((GUB - GLB) \cdot 100)$ , all in percent.

### Discussion of the results when extending the run time limit

The results for the re-runs with extended time limit do, in general, not indicate any considerable gain from having information well in advance of planning the container positionings. The improvements in lower and upper bounds for the optimal solution vary between 3.09 % and 12.28 %, resulting in reductions in GAP by up to about 50 %. Reducing the optimality gap by 51.08 %, which is the case when running the 10M1.s instance by the CPP model, is somewhat an acceptable improvement but in this case, the solution value is improved by only 5.88 %, suggesting that no overall advantage can be expected when allowing longer run times.

Considering that changing the time limit from 1,800 seconds to 28,800 seconds is a significant extension, corresponding to increasing the run time by a factor of 16, it must be concluded that the gain does not bear comparison with the effort and, thus, no attempts to access information several hours before handling the containers can be recommended if it involves any considerable effort or expences.

#### 5.1.4 Results for the models with relaxed lifo constraints

An investigation of the potential advantage of applying a cutting-plane approach is performed by omitting the lifo constraints and running the medium-scale class problems in the scaled versions. Appendix G contains all solutions from these runs. The purpose is to test whether there is a significant gain in run time, indicating that applying a separation routine and adding violated constraints ad hoc may be an advantageous direction. In order for this approach to be worthwhile exploring, there must be a considerable reduction in run times and the solutions must be somewhat close to feasible, only violating a minority of the omitted constraints.

Tables 5.11 and 5.12 contain the results from the relaxed CPP and CPPT model respectively. Each row represents a scaled problem instance from the medium-scale class and four sets of columns compare the relaxed runs with the original full model runs. The **Solution<sup>orig</sup>** columns repeat the results for the original instance run and the **Solution<sup>lifo</sup>** columns state results for the relaxed model. Equivalently, the **Run<sup>orig</sup>** and **Run<sup>lifo</sup>** columns represent the characteristics for the original and the relaxed run respectively. Keys to each of the columns are equal to previous solution tables in this section and can be reread in table 5.1 on page 109.

| Medium<br>scaled | Solution <sup>orig</sup> |    |       |  | Solution <sup>lifo</sup> |    |      | Run <sup>orig</sup> |                  |       | Run <sup>lifo</sup> |                  |      |
|------------------|--------------------------|----|-------|--|--------------------------|----|------|---------------------|------------------|-------|---------------------|------------------|------|
|                  | OPT                      | #M | GAP   |  | OPT                      | #M | GAP  | #N                  | CPU <sup>1</sup> | CPU*  | #N                  | CPU <sup>1</sup> | CPU* |
| 6S.s             | [25.08,31]               | 9  | 23.62 |  | 18                       | 6  | 0.00 | 479,419             | 195.6            | -     | 1                   | 0.1              | 0.3  |
| 6C.s             | 18                       | 6  | 0.00  |  | 18                       | 6  | 0.00 | 1                   | 0.1              | 0.5   | 1                   | 0.1              | 0.3  |
| 6M1.s            | 27                       | 7  | 0.00  |  | 18                       | 6  | 0.00 | 97,993              | 16.1             | 504.2 | 1                   | 0.1              | 0.4  |
| 6M2.s            | 26                       | 6  | 0.00  |  | 18                       | 6  | 0.00 | 19,765              | 0.1              | 103.1 | 1                   | 0.1              | 0.3  |
| 6M3.s            | 24                       | 6  | 0.00  |  | 18                       | 6  | 0.00 | 5,637               | 0.1              | 36.6  | 1                   | 0.0              | 0.3  |
| 6M4.s            | 24                       | 6  | 0.00  |  | 18                       | 6  | 0.00 | 11,163              | 0.1              | 61.0  | 1                   | 0.0              | 0.3  |
| 6M5.s            | 24                       | 6  | 0.00  |  | 18                       | 6  | 0.00 | 18,053              | 5.7              | 95.3  | 1                   | 0.0              | 0.3  |
| 6M6.s            | 22                       | 6  | 0.00  |  | 18                       | 6  | 0.00 | 1,515               | 0.1              | 12.3  | 1                   | 0.0              | 0.3  |
| 8S.s             | [34.00,42]               | 12 | 23.53 |  | 32                       | 8  | 0.00 | 226,544             | 150.4            | -     | 313                 | 0.1              | 4.4  |
| 8C.s             | 32                       | 8  | 0.00  |  | 32                       | 8  | 0.00 | 363                 | 0.2              | 17.9  | 287                 | 0.1              | 4.3  |
| 8M1.s            | [34.00,38]               | 8  | 11.76 |  | 32                       | 8  | 0.00 | 246,233             | 1.1              | -     | 275                 | 0.1              | 4.0  |
| 8M2.s            | [34.00,38]               | 10 | 11.76 |  | 32                       | 8  | 0.00 | 242,766             | 1,785.4          | -     | 299                 | 0.1              | 4.2  |
| 8M3.s            | 35                       | 9  | 0.00  |  | 32                       | 8  | 0.00 | 81,860              | 0.3              | 649.7 | 293                 | 0.1              | 4.2  |
| 8M4.s            | 35                       | 9  | 0.00  |  | 32                       | 8  | 0.00 | 73,775              | 240.5            | 625.7 | 287                 | 0.1              | 4.1  |
| 8M5.s            | 33                       | 9  | 0.00  |  | 32                       | 8  | 0.00 | 5,080               | 75.7             | 82.2  | 319                 | 0.1              | 4.3  |
| 8M6.s            | [33.06,36]               | 8  | 8.88  |  | 32                       | 8  | 0.00 | 247,403             | 126.3            | -     | 301                 | 0.1              | 4.4  |
| 8M7.s            | 34                       | 8  | 0.00  |  | 32                       | 8  | 0.00 | 40,653              | 28.9             | 324.2 | 307                 | 0.1              | 4.3  |
| 8M8.s            | 34                       | 8  | 0.00  |  | 32                       | 8  | 0.00 | 34,223              | 2.8              | 295.3 | 427                 | 0.1              | 5.3  |
| 10S.s            | [40.48,60]               | 18 | 48.24 |  | 40                       | 10 | 0.00 | 147,141             | 1,397.5          | -     | 1,031               | 0.1              | 13.9 |
| 10C.s            | 40                       | 10 | 0.00  |  | 40                       | 10 | 0.00 | 1,121               | 0.3              | 62.0  | 1,023               | 0.1              | 13.7 |
| 10M1.s           | [40.78,51]               | 11 | 25.07 |  | 40                       | 10 | 0.00 | 119,549             | 1,134.7          | -     | 1,155               | 0.1              | 15.5 |
| 10M2.s           | [40.72,54]               | 12 | 32.61 |  | 40                       | 10 | 0.00 | 147,060             | 88.3             | -     | 965                 | 0.1              | 14.0 |
| 10M3.s           | [41.00,46]               | 10 | 12.20 |  | 40                       | 10 | 0.00 | 164,271             | 549.6            | -     | 943                 | 0.1              | 12.7 |
| 10M4.s           | [40.81,46]               | 10 | 12.71 |  | 40                       | 10 | 0.00 | 182,239             | 523.7            | -     | 1,245               | 0.1              | 15.2 |
| 10M5.s           | [41.00,46]               | 12 | 12.20 |  | 40                       | 10 | 0.00 | 109,550             | 815.9            | -     | 1,369               | 0.1              | 18.4 |
| 10M6.s           | [41.00,45]               | 11 | 9.76  |  | 40                       | 10 | 0.00 | 182,701             | 351.2            | -     | 1,123               | 0.1              | 14.2 |
| 10M7.s           | [41.00,45]               | 11 | 9.76  |  | 40                       | 10 | 0.00 | 152,216             | 571.4            | -     | 1,269               | 0.1              | 15.5 |
| 10M8.s           | [40.38,49]               | 11 | 21.35 |  | 40                       | 10 | 0.00 | 124,198             | 184.9            | -     | 1,095               | 0.1              | 14.3 |
| 10M9.s           | [41.00,44]               | 12 | 7.32  |  | 40                       | 10 | 0.00 | 186,641             | 36.7             | -     | 1,225               | 0.1              | 15.2 |
| 10M10.s          | [41.00,44]               | 12 | 7.32  |  | 40                       | 10 | 0.00 | 193,173             | 264.6            | -     | 1,101               | 0.1              | 14.6 |

Table 5.11: Results for the medium-scale class problems in scaled versions run by the CPP model without the lifo constraints.

| Medium<br>scaled | Solution <sup>orig</sup> |    |        |  | Solution <sup>lifo</sup> |    |        | Run <sup>orig</sup> |                  |         | Run <sup>lifo</sup> |                  |       |
|------------------|--------------------------|----|--------|--|--------------------------|----|--------|---------------------|------------------|---------|---------------------|------------------|-------|
|                  | OPT                      | #M | GAP    |  | OPT                      | #M | GAP    | #N                  | CPU <sup>1</sup> | CPU*    | #N                  | CPU <sup>1</sup> | CPU*  |
| 6S.s             | [22.11,45.00]            | 15 | 103.54 |  | 24                       | 6  | 0.00   | 1                   | 617.8            | -       | 1                   | 146.5            | 146.5 |
| 6C.s             | [21.76,32.00]            | 10 | 47.09  |  | 24                       | 6  | 0.00   | 1                   | 254.4            | -       | 8                   | 201.9            | 202.0 |
| 6M1.s            | [24.00,42.00]            | 14 | 75.00  |  | 24                       | 6  | 0.00   | 3                   | 1,659.5          | -       | 1                   | 68.3             | 68.4  |
| 6M2.s            | [24.00,34.00]            | 10 | 41.67  |  | 24                       | 6  | 0.00   | 4                   | 1,669.9          | -       | 1                   | 126.6            | 126.7 |
| 6M3.s            | [24.00,38.00]            | 12 | 58.33  |  | 24                       | 6  | 0.00   | 26                  | 978.6            | -       | 1                   | 40.6             | 49.5  |
| 6M4.s            | [24.00,42.00]            | 14 | 75.00  |  | 24                       | 6  | 0.00   | 1                   | 141.5            | -       | 7                   | 104.2            | 104.3 |
| 6M5.s            | 24                       | 6  | 0.00   |  | 24                       | 6  | 0.00   | 3                   | 1,235.4          | 1,235.6 | 3                   | 123.0            | 123.1 |
| 6M6.s            | 24                       | 6  | 0.00   |  | 24                       | 6  | 0.00   | 6                   | 1,743.9          | 1,744.1 | 1                   | 19.1             | 80.0  |
| 8S.s             | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (c)     | -                   | -                | (e)   |
| 8C.s             | -                        | -  | -      |  | [24.33,55]               | 23 | 126.01 | -                   | -                | (e)     | 1                   | 706.1            | -     |
| 8M1.s            | -                        | -  | -      |  | [30.33,69]               | 25 | 127.53 | -                   | -                | (d)     | 1                   | 1,075.5          | -     |
| 8M2.s            | -                        | -  | -      |  | [24.35,60]               | 24 | 146.40 | -                   | -                | (d)     | 1                   | 1,034.1          | -     |
| 8M3.s            | -                        | -  | -      |  | [33.33,62]               | 22 | 86.04  | -                   | -                | (d)     | 1                   | 813.3            | -     |
| 8M4.s            | -                        | -  | -      |  | [34.00,59]               | 19 | 73.53  | -                   | -                | (e)     | 1                   | 1,797.7          | -     |
| 8M5.s            | -                        | -  | -      |  | [34.00,49]               | 19 | 44.12  | -                   | -                | (e)     | 1                   | 518.2            | -     |
| 8M6.s            | -                        | -  | -      |  | [31.37,47]               | 15 | 49.81  | -                   | -                | (e)     | 1                   | 660.3            | -     |
| 8M7.s            | -                        | -  | -      |  | [33.21,58]               | 24 | 74.65  | -                   | -                | (e)     | 1                   | 518.5            | -     |
| 8M8.s            | -                        | -  | -      |  | [24.36,54]               | 22 | 121.66 | -                   | -                | (e)     | 1                   | 592.3            | -     |
| 10S.s            | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |
| 10C.s            | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |
| 10M1.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (b)     | -                   | -                | (e)   |
| 10M2.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (b)     | -                   | -                | (e)   |
| 10M3.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (b)     | -                   | -                | (e)   |
| 10M4.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |
| 10M5.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (b)     | -                   | -                | (e)   |
| 10M6.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |
| 10M7.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (b)     | -                   | -                | (e)   |
| 10M8.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |
| 10M9.s           | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |
| 10M10.s          | -                        | -  | -      |  | -                        | -  | -      | -                   | -                | (a)     | -                   | -                | (e)   |

Table 5.12: Results for the medium-scale class problems in scaled versions run by the CPPT model without the lifo constraints. (a) Out of memory while generating the matrix. (b) Out of memory while loading the problem. (c) Out of memory during pre-solve. (d) Out of memory during optimization. (e) Exceeding time limit while solving LP relaxation.



## Discussion of the results when omitting lifo constraints

The results for the relaxed CPP runs, displayed in table 5.11, indicate a significant gain in closing the gap as well as in run times by omitting the lifo constraints. All the problem instances are solved to optimality in few seconds: Less than 1.0 second for the 6 containers subclass, around five seconds for the 8 containers subclass, and not more than 20 seconds for the 10 containers instances.

However, the results for the CPPT model without the lifo constraints, gathered in table 5.12, do not suggest the same advantage of the relaxation approach. In the 6 containers subclass, all instances are solved to optimality within the time limit, solution times varying between one and three minutes, which is the case for only two of the problems when running the full CPPT model. Feasible solutions to all 8 containers problems except for the stair instance are found within the time limit but the gap is not closed in any of the cases. Even though this is an improvement to the full model runs - exceeding the time limit when solving the root node in six instances and running out of memory during pre-solve or optimization in the remaining four instances - the performance is quite poor. A similar pattern shows in the 10 containers subclass, exceeding the time limit while solving the root node when omitting lifo constraints compared to running out of memory even before pre-solving in the full CPPT model runs.

Furthermore, by investigation of the solutions for the relaxed model runs in appendix G, it is clear that the trivial solution with all containers stacked without consideration for time conflicts is found in all cases. In the CPP solutions, as no capacity constraints prevent the tall stacks, all containers are placed at the position with the shortest transportation time from the arrival place and/or to the departure place. In the CPPT solutions,  $H$  containers are placed at the best position, another  $H$  at the second best position, and so on. This means that close to the worst case number of time conflicts,  $\sum_{i=1}^{C-1} i$ , occur in all solutions except for the cone instances. This issue is not unexpected as the lifo constraints tie the problem together across containers' individual paths through the block network. Violation of a large number of lifo restrictions implies that almost all relaxed constraints must be added to reach feasibility.

Given the poor performance of the relaxed model runs as well as the far from feasible solutions, it must be concluded that applying a cutting-plane approach, consisting in relaxation of the lifo restrictions and adding violated constraints detected by a separation routine, is not a fruitful direction.

### 5.1.5 Conclusive remarks on the model results

The results presented in this section lead to some general conclusions on the model approach to solving the problem considered in this study. First, results from the validation and evaluation phases, solving the small- and medium-scale class problems by each of the models, using the Xpress-MP solver, have been presented and next, two approaches to solving the models on alternative terms compared to the main solution procedure have been investigated.

Comparing the CPP and CPPT model runs on the tight small-scale test problems

leaves the impression that the CPPT model performs better in terms of run time and the need for branching but when considering the scaled instances, the picture changes as CPPT model run times increase significantly and several instances from the 5 containers subclass are not solved to optimality within the time limit of 1,800 seconds. This observation is confirmed when comparing the models in the medium-scale test phase where the CPPT model performs slightly better on average when solving the tight 6 and 8 containers instances but is outperformed by the CPP model when increasing the problem size to 10 containers and - even more so - when extending the time horizon.

The increase in optimality gap as well as the poor performance concerning run time and complexity of the problem matrix, causing the model to run out of memory while generating or solving the scaled 8 and 10 containers instances, suggest that the CPPT model is not an improvement to the CPP model when solving problems of increasing size. However, we recall that the CPP model does not include capacity constraints on the stacking height and the number of simultaneous moves, making the CPPT model more true to real-life restrictions but also requiring more computational work when searching the branch and bound tree.

Another aspect of the model performances concerns the root node quality. In general, the CPPT model yields a better LP relaxation than the CPP model, resulting in better root node bounds which is evidently stated by the poor CPP QUAL values in contrast with the generally positive or close to 100 % quality of the CPPT root node, at first suggesting that optimum can be reached fastest by the CPPT model. When this is not the case, except for the very smallest instances, the explanation may concern the size of the branch and bound tree as well as the time required to solve each node. The number of variables and constraints in the CPPT model increases considerably when scaling the problem, especially the time dimension, leading to a large number of nodes to branch on as well as extensive calculation requirements in each node. Therefore, in spite of the good initial bound(s) found in the root node, RLB and RUB, leaving a smaller gap to close, searching the tree nodes requires far more computational power when solving the CPPT model compared to solving the CPP model.

Figure 5.4 illustrates the concept of the quality of the root node solution compared to the efficiency in finding the global optimal solution. Even though the initial CPPT lower bound provided by the LP solution, and possibly the first feasible solution found in the root node, most often outperform the solution of the CPP root node, the size of the CPPT tree implies a much larger number of solutions to investigate and the increased number of constraints implies longer solution times at each node.

Concluding that neither the CPP nor the CPPT model is suited for solving larger problem instances, two alternative approaches - extending the upper bound on the run time and relaxing the lifo constraints - have been investigated. The gain in solution quality when allowing longer run times is far from comparable with the extent to which the time limit is increased or with the effort, presumably required in order to obtain the information several hours in advance of handling the containers. Results from the model runs when omitting the lifo constraints suggest a considerable gain concerning run time and optimality gap for the CPP model but not the same advantage when relaxing the

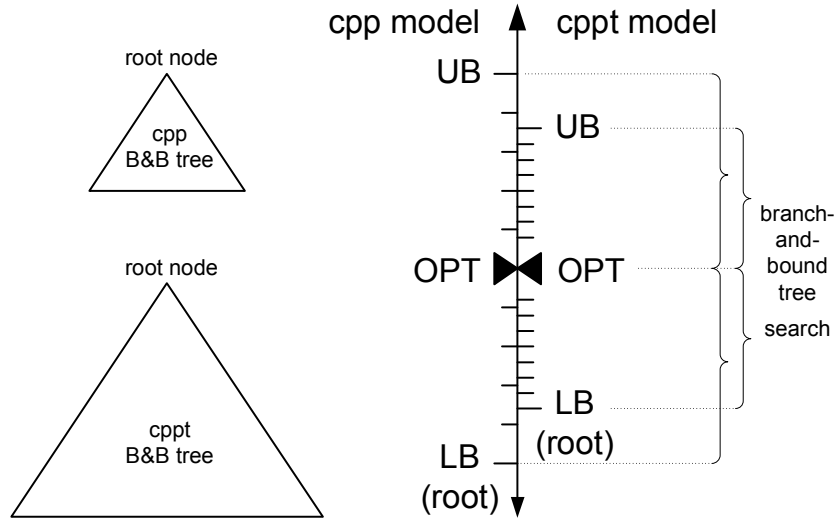


Figure 5.4: Conceptual difference between the CPP and CPPT models, concerning the quality of the root node solution and the subsequent search for the optimal solution. In spite of good initial bound(s) in the root node, searching the CPPT branch and bound tree is far more time-consuming than for the CPP model due to a larger tree as well as extensive computations in each node.

CPPT model. In addition to the uncertain improvement in the models' performance, the solution structure falls apart when omitting the lifo constraints, resulting in trivial "one container at a time" solutions with a large number of constraint violations due to the disregard for time conflicts.

Concluding, the approaches to solve the models using the Xpress-MP solver, investigated in this research study, are not suited for handling problems of considerable sizes. Therefore, alternative solution methods must be explored. One direction is further development of an exact approach, constructing a problem-specific algorithm with tailored procedures as alternative to the standard routines performed by Xpress-MP or other standard solvers. Generally, research experience shows that run times may be reduced significantly when using an algorithm, specially designed to solve a specific problem by a certain model and/or particular routines. Alternatively, solving the problem by a heuristic algorithm may be a suitable approach, especially if considering industrial application where reaching optimum is less relevant than finding good solutions within short run times. The results for the heuristic, described in chapter 3, are presented in the following section and conclusions on the different solution approaches, their performance and relevance, are summarized in section 5.3.

## 5.2 Results for the heuristic algorithm

This section reports the computational results for the heuristic algorithm. Results for the small- and medium-scale class problems, tight as well as scaled instances, serve partly as parameter tuning and partly as validation of the algorithm. The tight instance solutions can be compared to the models and results from the the scaled instances can be compared to the model solutions as regards solution time and the objective concerning transportation time and full moves. Results for the large-scale instances demonstrate the quality and efficiency of the heuristic algorithm as well as improvement routines when handling real-life problems.

In each small- and medium-scale subclass, the results for the tight and the scaled instances are presented separately. Each row starting with the problem instance name - e.g. **3S.t** for the tight version of the 3 containers stair problem, **5C.s** for the scaled version of the 5 containers cone problem, and **8M1.s** - **8M8.s** for the scaled versions of the 8 containers mix problem instances - states the results for the heuristic algorithm, allowing rejection of containers, before applying the improvement routine. Rows starting with **-no R** represent results for the heuristic algorithm where rejections are not accepted. For the scaled problem instances, a third row for each instance, starting with **-shift**, shows the effect of the improvement routine, shifting late containers. However, due the good results for the basic heuristic, implying that the  $\alpha$  parameters are based on well-chosen  $\eta$  values, runs with poorer tuning parameters are required to illustrate the effect of the shifting routine. Results from these runs are displayed together with the other tuning results in appendix H and two tables, 5.22 with the 8 containers subclass and 5.25 with the 10 containers subclass, containing instances where the effect of shifting late containers is captured, are showed and commented in this section.

In each large-scale subclass, the results are presented according to the search strategy,  $\rho = 25\%$ ,  $\rho = 33\%$ , ...,  $\rho = 100\%$  of  $P$ . Rows starting with the problem instance name, representing the number of containers  $C$ , the number of positions  $P$ , and the  $R/B$  ratio in percent, contain results for the heuristic algorithm when allowing rejection of containers and, equivalent to the scaled versions of the small- and medium-scale problems, rows starting with **-no R** and **-shift** state the results when no rejections are accepted and after applying the improvement routine respectively. In these rows, a - is stated when the value is identical to the above table cell.

Solutions to all heuristic runs, excluding the tuning phase, are reported in appendix I. In the large-scale class where two different search strategies are investigated, the solutions for the combined forward and backwards search are displayed.

In the result tables 5.14 - 5.75, six **Solution** columns represent the total transportation time  $T$ , the number of full and empty moves  $\#M$ , the extend to which containers wait at the arrival place, are early or late at the departure place with both the number of containers and the total time stated, and the number of containers and percentage of  $C$  rejected. In the four **Criteria** columns, the number of moves performed by choosing positions by criterion 0, 1, 2, and 3 are given. Finally, the **Performance** columns contain a lower bound on the number of full moves,  $LB = 2C$ , the gap between this bound and

the number of full moves in the solution, the quality of the solution based on the number of full moves and LB, and the total CPU time. Note that  $LB = 2C$  may be very poor, especially for instances where reshuffling is inevitable, and that QUAL does not consider punctuality, i.e. it is not affected by waiting time or containers being early or late for departure. The GAP and QUAL columns, therefore, should be read with this in mind. Keys to the solution tables are summarized in table 5.13.

| <b>Solution</b>    |                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| T                  | Total transportation time in a best found solution.                                                                                     |
| #M                 | Total number of full/empty moves in a best found solution. In tight small- and medium-scale instances, the number of empty moves are 0. |
| Wait               | Total number/time of containers waiting at the arrival place before being moved to the first position.                                  |
| Early              | Total number/time of containers early at the departure place.                                                                           |
| Late               | Total number/time of containers late at the departure place.                                                                            |
| R                  | Total number/percentage of containers rejected due to lack of feasible positions.                                                       |
| <b>Criteria</b>    |                                                                                                                                         |
| 0                  | Total number of moves performed by criterion 0, i.e. moves to an empty position.                                                        |
| 1                  | Total number of moves performed by criterion 1, i.e. moves to a position where the container on top has the same departure time.        |
| 2                  | Total number of moves performed by criterion 2, i.e. moves to a position where the container on top departs after the moved one.        |
| 3                  | Total number of moves performed by criterion 3, i.e. moves to a position where the container on top departs before the moved one.       |
| <b>Performance</b> |                                                                                                                                         |
| LB                 | Lower bound on the number of full moves, $LB = 2C$ .                                                                                    |
| GAP                | Gap in percent, $GAP = (\#M - LB)/LB \cdot 100$ , where #M is the number of full moves.                                                 |
| QUAL               | Quality of the solution in percent, $QUAL = 100 - (\#M - LB)/LB \cdot 100$ .                                                            |
| CPU                | Total CPU time in seconds.                                                                                                              |

Table 5.13: Key to heuristic solution tables 5.14 - 5.75.

### 5.2.1 Tuning phase

Tuning of the  $\alpha$  parameters is based on the default values, computed by the minimal quality function values  $\eta^{dist} = 0.75$ ,  $\eta_k^{time} = 0.25, 1.00, 0.75, 0.10$ , and  $\eta^{stack} = 0.75$  as described in section 3.1.3 on page 68. By examining the solutions from the first tuning, the gain from up- and/or down-scaling one or several parameters is estimated and re-runs are performed.

In general, the default values of  $\alpha$  lead to quite good results and in cases with room for improvement, it is evident that small adjustments may improve the solution. In the typical case, undesirable moves are performed by choice of positions by criterion 3, i.e. moves to a position where the top container is departing before the positioned one, leading to possibly redundant reshuffling at a later point in time. The reason for this is obviously too small values of  $\alpha_3^{time}$  but, in correlation with this, too large values of  $\alpha^{dist}$ ,  $\alpha_2^{time}$ , and  $\alpha^{stack}$  contribute to avoiding good positions due to unnecessary punishment of moves to more distant positions, to stacks with several containers in it, or to positions where there is a large, but positive, value of  $D_{c'} - D_c$ ,  $c'$  being the top container in the stack. Even though seeking to prevent moves over long distances, avoid uneven stack heights, and reserve certain positions for future containers that may require a stack where the top container departs relatively late, too large values of  $\alpha^{dist}$ ,  $\alpha_2^{time}$ , and  $\alpha^{stack}$ , combined with  $\alpha_3^{time}$  being too small, show to imply unwanted moves.

Consequently, adjustment of the  $\alpha$  values proceeds as follows. In each iteration,  $\alpha_0^{time}$  and  $\alpha_3^{time}$  are up-scaled as investigation of the solutions shows that too many positions are chosen by criterion 0, i.e. using an empty position for a container which could have been stacked on top of another one without time conflicting, thus reserving the empty position for a future container with a later departure time, and criterion 3, i.e. possibly causing additional reshuffling, sometimes even directly after being placed at the poor position. On the other hand,  $\alpha^{dist}$ ,  $\alpha_2^{time}$ , and  $\alpha^{stack}$  are down-scaled as too few positions are chosen by criterion 2, basically ensuring few reshuffles, or too many moves are “forced” by large values of  $\alpha^{dist}$  or  $\alpha^{stack}$  to avoid good positions either because of the distance or the stack height, even though they are better than alternative close or empty ones.

Starting from the default values, tuning  $n$  applies the following  $\alpha$  values,  $n - 1$  corresponding to the previous tuning phase:  $\alpha^{dist}(n) = 0.5 \alpha^{dist}(n - 1)$ ,  $\alpha_0^{time}(n) = 2 \alpha_0^{time}(n - 1)$ ,  $\alpha_2^{time}(n) = 0.5 \alpha_2^{time}(n - 1)$ ,  $\alpha_3^{time}(n) = 2 \alpha_3^{time}(n - 1)$ , and  $\alpha^{stack}(n) = 0.5 \alpha^{stack}(n - 1)$ . Results from the parameter tuning can be found in appendix H.

### 5.2.2 Small- and medium-scale class results

This section contains the results from the small- and medium-scale test phases. Each table represents either tight or scaled instances. We recall that for the tight instances, the total transportation time  $T$  and number of moves  $\#M$  can be compared directly to the objective value of the CPPT model, adjusted by  $\#M$  and  $C$  respectively as described on page 108. In addition to the result tables, some interesting outcomes of the tuning phase with poor  $\alpha$  values (Tuning 0) are presented in tables 5.22 and 5.25, thus illustrating the improvement of the initial heuristic solution by the shifting routine. Solutions to small- and medium-scale class problems can be found in appendices I.1 and I.2.

| $C = 3$ |    | Solution |       |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| tight   | T  | #M       | Wait  | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.t    | 10 | 7/0      | 1/1.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 10 | 7/0      | 1/1.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| 3C.t    | 8  | 6/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8  | 6/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M1A.t  | 10 | 6/0      | 1/1.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 10 | 6/0      | 1/1.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M1B.t  | 10 | 6/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 10 | 6/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M2A.t  | 8  | 6/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8  | 6/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M2B.t  | 8  | 6/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8  | 6/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M3A.t  | 8  | 6/0      | 1/1.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8  | 6/0      | 1/1.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M3B.t  | 8  | 6/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8  | 6/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |

Table 5.14: Results for the 3 containers tight problem instances.

| $C = 3$ |    | Solution |       |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| scaled  | T  | #M       | Wait  | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.s    | 18 | 7/6      | 2/5.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 18 | 7/6      | 2/5.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| - shift | 18 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3C.s    | 12 | 6/4      | 2/3.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 12 | 6/4      | 2/3.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 12 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1A.s  | 18 | 6/5      | 2/5.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 18 | 6/5      | 2/5.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 18 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1B.s  | 18 | 6/5      | 1/2.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 18 | 6/5      | 1/2.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 18 | -        | -     | 0/0.0 | 1/1.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2A.s  | 14 | 6/5      | 2/3.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14 | 6/5      | 2/3.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2B.s  | 14 | 6/5      | 2/3.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14 | 6/5      | 2/3.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3A.s  | 14 | 6/4      | 2/5.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14 | 6/4      | 2/5.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3B.s  | 14 | 6/4      | 1/2.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14 | 6/4      | 1/2.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14 | -        | -     | 0/0.0 | 1/1.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.15: Results for the 3 containers scaled problem instances.



| $C = 4$ |    | Solution |       |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| tight   | T  | #M       | Wait  | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.t    | 14 | 10/0     | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 14 | 10/0     | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| 4C.t    | 10 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 10 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M1A.t  | 14 | 9/0      | 0/0.0 | 0/0.0 | 2/2.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 14 | 9/0      | 0/0.0 | 0/0.0 | 2/2.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M1B.t  | 14 | 9/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 14 | 9/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M2A.t  | 12 | 9/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 12 | 9/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M2B.t  | 12 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 12 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M3A.t  | 12 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 12 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M3B.t  | 10 | 8/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 10 | 8/0      | 0/0.0 | 0/0.0 | 1/1.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M4A.t  | 12 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 12 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M4B.t  | 10 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 10 | 8/0      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |

Table 5.16: Results for the 4 containers tight problem instances.

| $C = 4$ |    | Solution |       |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| scaled  | T  | #M       | Wait  | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.s    | 26 | 10/9     | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - no R  | 26 | 10/9     | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - shift | 26 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4C.s    | 16 | 8/6      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 16 | 8/6      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 16 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1A.s  | 26 | 9/8      | 2/6.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 26 | 9/8      | 2/6.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 26 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1B.s  | 26 | 9/8      | 2/6.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 26 | 9/8      | 2/6.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 26 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2A.s  | 22 | 9/8      | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 22 | 9/8      | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 22 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2B.s  | 22 | 8/7      | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 22 | 8/7      | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3A.s  | 22 | 8/7      | 2/4.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 22 | 8/7      | 2/4.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3B.s  | 18 | 8/7      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 18 | 8/7      | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4A.s  | 22 | 8/6      | 2/6.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 22 | 8/6      | 2/6.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4B.s  | 18 | 8/6      | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 18 | 8/6      | 1/2.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18 | -        | -     | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.17: Results for the 4 containers scaled problem instances.

| $C = 5$ |    | Solution |       |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| tight   | T  | #M       | Wait  | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.t    | 18 | 12/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 18 | 12/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.0 |
| 5C.t    | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M1A.t  | 18 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 18 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M1B.t  | 18 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 18 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M2A.t  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M2B.t  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M3A.t  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M3B.t  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M4A.t  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M4B.t  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M5A.t  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M5B.t  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14 | 10/0     | 0/0.0 | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |

Table 5.18: Results for the 5 containers tight problem instances.

| C = 5   |    | Solution |        |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|--------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| scaled  | T  | #M       | Wait   | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.s    | 34 | 12/11    | 1/2.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 34 | 12/11    | 1/2.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - shift | 34 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5C.s    | 24 | 10/8     | 1/2.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 24 | 10/8     | 1/2.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 24 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1A.s  | 34 | 10/9     | 3/12.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 34 | 10/9     | 3/12.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 34 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1B.s  | 34 | 10/9     | 3/12.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 34 | 10/9     | 3/12.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 34 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2A.s  | 30 | 10/9     | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30 | 10/9     | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2B.s  | 30 | 10/9     | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30 | 10/9     | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3A.s  | 30 | 10/9     | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30 | 10/9     | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3B.s  | 30 | 10/9     | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30 | 10/9     | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4A.s  | 30 | 10/9     | 3/10.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30 | 10/9     | 3/10.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4B.s  | 26 | 10/9     | 2/4.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26 | 10/9     | 2/4.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5A.s  | 26 | 10/8     | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 26 | 10/8     | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5B.s  | 26 | 10/8     | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26 | 10/8     | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.19: Results for the 5 containers scaled problem instances.

| $C = 6$ |    | Solution |       |        |       |        | Criteria |   |   |   | Performance |        |        |     |
|---------|----|----------|-------|--------|-------|--------|----------|---|---|---|-------------|--------|--------|-----|
| tight   | T  | #M       | Wait  | Early  | Late  | R      | 0        | 1 | 2 | 3 | LB          | GAP    | QUAL   | CPU |
| 6S.t    | 22 | 15/0     | 0/0.0 | 0/0.0  | 1/1.0 | 0/0.0  | 4        | 0 | 2 | 3 | 12          | 25.00  | 75.00  | 0.1 |
| - no R  | 22 | 15/0     | 0/0.0 | 0/0.0  | 1/1.0 | 0/0.0  | 4        | 0 | 2 | 3 | 12          | 25.00  | 75.00  | 0.0 |
| 6C.t    | 18 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M1.t   | 22 | 13/0     | 0/0.0 | 0/0.0  | 2/2.0 | 0/0.0  | 4        | 0 | 2 | 1 | 12          | 8.33   | 91.67  | 0.1 |
| - no R  | 22 | 13/0     | 0/0.0 | 0/0.0  | 2/2.0 | 0/0.0  | 4        | 0 | 2 | 1 | 12          | 8.33   | 91.67  | 0.0 |
| 6M2.t   | 20 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| - no R  | 20 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M3.t   | 18 | 12/0     | 0/0.0 | 0/0.0  | 2/2.0 | 0/0.0  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18 | 12/0     | 0/0.0 | 0/0.0  | 2/2.0 | 0/0.0  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M4.t   | 18 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M5.t   | 18 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| - no R  | 18 | 12/0     | 0/0.0 | 0/0.0  | 0/0.0 | 0/0.0  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M6.t   | 14 | 10/0     | 0/0.0 | 1/14.0 | 0/0.0 | 1/16.7 | 2        | 0 | 3 | 0 | 12          | -16.67 | 116.67 | 0.1 |
| - no R  | 18 | 12/0     | 0/0.0 | 0/0.0  | 1/1.0 | 0/0.0  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |

Table 5.20: Results for the 6 containers tight problem instances.

| $C = 6$ |    | Solution |        |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|--------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| scaled  | T  | #M       | Wait   | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 6S.s    | 42 | 15/14    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.0 |
| - no R  | 42 | 15/14    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.1 |
| - shift | 42 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6C.s    | 32 | 12/10    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 32 | 12/10    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M1.s   | 42 | 13/12    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.0 |
| - no R  | 42 | 13/12    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.0 |
| - shift | 42 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M2.s   | 38 | 12/11    | 3/12.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - no R  | 38 | 12/11    | 3/12.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 38 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M3.s   | 34 | 12/11    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34 | 12/11    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M4.s   | 34 | 12/11    | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34 | 12/11    | 3/8.0  | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M5.s   | 34 | 12/11    | 3/10.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - no R  | 34 | 12/11    | 3/10.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M6.s   | 32 | 12/10    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - no R  | 32 | 12/10    | 2/6.0  | 0/0.0 | 0/0.0 | 0/0.0 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.21: Results for the 6 containers scaled problem instances.

| $C = 8$ |     | Solution |        |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|---------|-----|----------|--------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T   | #M       | Wait   | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.s    | 65  | 20/19    | 5/34.0 | 0/0.0 | 1/8.0  | 0/0.0 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - no R  | 65  | 20/19    | 5/34.0 | 0/0.0 | 1/8.0  | 0/0.0 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - shift | 65  | -        | -      | 0/0.0 | 1/8.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8C.s    | 55  | 16/14    | 5/28.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 55  | 16/14    | 5/28.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 55  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M1.s   | 65  | 17/16    | 6/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - no R  | 65  | 17/16    | 6/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - shift | 65  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M2.s   | 69  | 18/17    | 5/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - no R  | 69  | 18/17    | 5/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - shift | 69  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M3.s   | 61  | 16/15    | 6/36.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61  | 16/15    | 6/36.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M4.s   | 61  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M5.s   | 57  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 57  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M6.s   | 65  | 18/17    | 5/36.0 | 0/0.0 | 2/10.0 | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - no R  | 65  | 18/17    | 5/36.0 | 0/0.0 | 2/10.0 | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - shift | 67  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M7.s   | 55  | 16/15    | 5/26.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 55  | 16/15    | 5/26.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 55  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M8.s   | 55  | 16/14    | 5/24.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 55  | 16/14    | 5/24.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 55. | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.22: Tuning 0 on the 8 containers scaled problem instances. One container is late by 8 in the **8S.s** instance but it is not possible to improve the solution by the shifting routine. In the **8M6.s** instance, however, the shifting routine repairs the initial heuristic solution where two containers are late by 10. The improvement is performed in less than 0.1 seconds.

| $C = 8$ |    | Solution |       |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| tight   | T  | #M       | Wait  | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.t    | 32 | 20/0     | 0/0.0 | 0/0.0 | 6/15.0 | 0/0.0 | 4        | 0 | 4 | 4 | 16          | 25.00 | 75.00  | 0.1 |
| - no R  | 32 | 20/0     | 0/0.0 | 0/0.0 | 6/15.0 | 0/0.0 | 4        | 0 | 4 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| 8C.t    | 30 | 16/0     | 0/0.0 | 0/0.0 | 6/15.0 | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30 | 16/0     | 0/0.0 | 0/0.0 | 6/15.0 | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M1.t   | 32 | 16/0     | 2/2.0 | 0/0.0 | 2/4.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32 | 16/0     | 2/2.0 | 0/0.0 | 2/4.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M2.t   | 32 | 16/0     | 1/1.0 | 0/0.0 | 2/2.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32 | 16/0     | 1/1.0 | 0/0.0 | 2/2.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M3.t   | 32 | 16/0     | 0/0.0 | 0/0.0 | 5/13.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32 | 16/0     | 0/0.0 | 0/0.0 | 5/13.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M4.t   | 30 | 16/0     | 0/0.0 | 0/0.0 | 8/17.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 30 | 16/0     | 0/0.0 | 0/0.0 | 8/17.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M5.t   | 30 | 16/0     | 1/1.0 | 0/0.0 | 8/17.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 30 | 16/0     | 1/1.0 | 0/0.0 | 8/17.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M6.t   | 30 | 16/0     | 0/0.0 | 0/0.0 | 7/15.0 | 0/0.0 | 4        | 0 | 3 | 1 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 30 | 16/0     | 0/0.0 | 0/0.0 | 7/15.0 | 0/0.0 | 4        | 0 | 3 | 1 | 16          | 0.00  | 100.00 | 0.0 |
| 8M7.t   | 32 | 16/0     | 0/0.0 | 0/0.0 | 8/20.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32 | 16/0     | 0/0.0 | 0/0.0 | 8/20.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M8.t   | 28 | 16/0     | 0/0.0 | 0/0.0 | 3/5.0  | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 28 | 16/0     | 0/0.0 | 0/0.0 | 3/5.0  | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |

Table 5.23: Results for the 8 containers tight problem instances.



| $C = 8$ |    | Solution |        |       |       |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|--------|-------|-------|-------|----------|---|---|---|-------------|-------|--------|-----|
| scaled  | T  | #M       | Wait   | Early | Late  | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.s    | 61 | 20/19    | 4/20.0 | 0/0.0 | 1/8.0 | 0/0.0 | 4        | 0 | 4 | 4 | 16          | 25.00 | 75.00  | 0.1 |
| - no R  | 61 | 20/19    | 4/20.0 | 0/0.0 | 1/8.0 | 0/0.0 | 4        | 0 | 4 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - shift | 61 | -        | -      | 0/0.0 | 1/8.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8C.s    | 57 | 16/14    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 57 | 16/14    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M1.s   | 61 | 16/15    | 6/46.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61 | 16/15    | 6/46.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M2.s   | 61 | 16/15    | 5/32.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61 | 16/15    | 5/32.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M3.s   | 61 | 16/15    | 6/32.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61 | 16/15    | 6/32.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M4.s   | 57 | 16/15    | 5/22.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 57 | 16/15    | 5/22.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M5.s   | 57 | 16/15    | 4/22.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 57 | 16/15    | 4/22.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M6.s   | 55 | 17/16    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - no R  | 55 | 17/16    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - shift | 55 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M7.s   | 61 | 16/15    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61 | 16/15    | 4/20.0 | 0/0.0 | 0/0.0 | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M8.s   | 53 | 16/14    | 5/24.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 53 | 16/14    | 5/24.0 | 0/0.0 | 0/0.0 | 0/0.0 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 53 | -        | -      | 0/0.0 | 0/0.0 | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.24: Results for the 8 containers scaled problem instances.

| C = 10  |    | Solution |       |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|-------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T  | #M       | Wait  | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.s   | 93 | 27/26    | 3/6.0 | 0/0.0 | 1/14.0 | 0/0.0 | 7        | 0 | 3 | 7 | 20          | 35.00 | 65.00  | 0.1 |
| - no R  | 93 | 27/26    | 3/6.0 | 0/0.0 | 1/14.0 | 0/0.0 | 7        | 0 | 3 | 7 | 20          | 35.00 | 65.00  | 0.0 |
| - shift | 95 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10C.s   | 69 | 20/18    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R  | 69 | 20/18    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift | 69 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M1.s  | 85 | 25/23    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 5 | 20          | 25.00 | 75.00  | 0.1 |
| - no R  | 85 | 25/23    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 5 | 20          | 25.00 | 75.00  | 0.0 |
| - shift | 85 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M2.s  | 81 | 23/22    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R  | 81 | 23/22    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift | 81 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M3.s  | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R  | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift | 77 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M4.s  | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R  | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift | 77 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M5.s  | 79 | 22/21    | 2/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R  | 79 | 22/21    | 2/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift | 79 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M6.s  | 79 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R  | 79 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift | 79 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M7.s  | 73 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R  | 73 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift | 73 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M8.s  | 77 | 21/20    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R  | 77 | 21/20    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 77 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M9.s  | 73 | 21/20    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R  | 73 | 21/20    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 73 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M10.s | 73 | 22/20    | 3/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R  | 73 | 22/20    | 3/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift | 73 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.25: Tuning 0 on the 10 containers scaled problem instances. The initial heuristic solution of the 10S.s instance delays one container by 10 which is fixed by the shifting routine, leaving no late containers after a run time of less than 0.1 seconds

| $C = 10$ |    | Solution |        |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|----------|----|----------|--------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| tight    | T  | #M       | Wait   | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.t    | 40 | 24/0     | 3/6.0  | 1/2.0 | 3/9.0  | 0/0.0 | 6        | 0 | 3 | 5 | 20          | 20.00 | 80.00  | 0.1 |
| - no R   | 40 | 24/0     | 3/6.0  | 1/2.0 | 3/9.0  | 0/0.0 | 6        | 0 | 3 | 5 | 20          | 20.00 | 80.00  | 0.0 |
| 10C.t    | 38 | 20/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38 | 20/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M1.t   | 38 | 20/0     | 1/2.0  | 0/0.0 | 1/2.0  | 0/0.0 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38 | 20/0     | 1/2.0  | 0/0.0 | 1/2.0  | 0/0.0 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M2.t   | 42 | 21/0     | 3/3.0  | 0/0.0 | 4/4.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 42 | 21/0     | 3/3.0  | 0/0.0 | 4/4.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M3.t   | 38 | 20/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38 | 20/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M4.t   | 38 | 21/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 38 | 21/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M5.t   | 40 | 22/0     | 4/15.0 | 0/0.0 | 4/10.0 | 0/0.0 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - no R   | 40 | 22/0     | 4/15.0 | 0/0.0 | 4/10.0 | 0/0.0 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| 10M6.t   | 40 | 21/0     | 1/4.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 40 | 21/0     | 1/4.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M7.t   | 40 | 20/0     | 1/2.0  | 0/0.0 | 1/2.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 40 | 20/0     | 1/2.0  | 0/0.0 | 1/2.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M8.t   | 38 | 21/0     | 0/0.0  | 0/0.0 | 1/1.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 38 | 21/0     | 0/0.0  | 0/0.0 | 1/1.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M9.t   | 40 | 20/0     | 0/0.0  | 0/0.0 | 1/1.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 40 | 20/0     | 0/0.0  | 0/0.0 | 1/1.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M10.t  | 38 | 20/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38 | 20/0     | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |

Table 5.26: Results for the 10 containers tight problem instances.

| C = 10  |    | Solution |        |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|---------|----|----------|--------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| scaled  | T  | #M       | Wait   | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.s   | 77 | 26/25    | 0/0.0  | 0/0.0 | 1/14.0 | 0/0.0 | 4        | 0 | 6 | 6 | 20          | 30.00 | 70.00  | 0.1 |
| - no R  | 77 | 26/25    | 0/0.0  | 0/0.0 | 1/14.0 | 0/0.0 | 4        | 0 | 6 | 6 | 20          | 30.00 | 70.00  | 0.0 |
| - shift | 79 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10C.s   | 73 | 20/18    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R  | 73 | 20/18    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift | 73 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M1.s  | 85 | 21/20    | 4/14.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R  | 85 | 21/20    | 4/14.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 85 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M2.s  | 89 | 23/22    | 3/12.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 6 | 2 | 20          | 15.00 | 85.00  | 0.0 |
| - no R  | 89 | 23/22    | 3/12.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 6 | 2 | 20          | 15.00 | 85.00  | 0.0 |
| - shift | 89 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M3.s  | 85 | 21/20    | 2/6.0  | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R  | 85 | 21/20    | 2/6.0  | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 85 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M4.s  | 77 | 21/20    | 1/2.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R  | 77 | 21/20    | 1/2.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 77 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M5.s  | 75 | 23/22    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - no R  | 75 | 23/22    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift | 75 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M6.s  | 73 | 22/21    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R  | 73 | 22/21    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift | 73 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M7.s  | 73 | 21/20    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R  | 73 | 21/20    | 0/0.0  | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 73 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M8.s  | 73 | 21/20    | 1/2.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R  | 73 | 21/20    | 1/2.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift | 73 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M9.s  | 77 | 20/19    | 2/4.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R  | 77 | 20/19    | 2/4.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift | 77 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M10.s | 73 | 20/18    | 3/8.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R  | 73 | 20/18    | 3/8.0  | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift | 73 | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table 5.27: Results for the 10 containers scaled problem instances.

## Discussion of the heuristic small- and medium-scale results

All small-scale problem instances, tight as well as scaled, are solved in 0.1 seconds or faster and there is a clear tendency towards the runs where rejections are not allowed and the shifting routine solving faster than the basic heuristic algorithm. Based on the computed lower bound on the number of full moves,  $LB = 2C$ , corresponding to a situation where all containers are positioned exactly once, the solution value  $\#M$  generally coincides, resulting in  $GAP = 0.00\%$  and  $QUAL = 100.00\%$ . All stair instances are exceptions to this performance level, the explanation lying in the lower bound which is clearly poor for these instances since some reshuffling is unavoidable for the given structure of arrival and departure times.

In these small and constructed problem instances, some degree of waiting and/or delay is likely to occur due to the density of arrival and departure times, leading to events most often being processed immediately after each other and, for the scaled instances' part, little time to perform the empty moves, e.g. required to return to the arrival place and pick up subsequent containers. When optimal model solutions let no containers wait before being placed in the block while the heuristic in some cases produce waiting time, it is due to the procedure by which positions are investigated when placing arrival containers. Searching positions from the first one available and forwards, the closest position is chosen if all candidates are equally good. Consequently, if e.g. the first three containers,  $c_1$ ,  $c_2$ , and  $c_3$ , arrive at  $A_1$ ,  $A_2$ , and  $A_3$ , where  $A_2 - A_1 > A_3 - A_2$ ,  $c_1$  is placed at the closest position  $p_1$  and, if departure times are not coincident or very close,  $c_2$  is placed at the second-closest position  $p_2$ , possibly leaving insufficient time to return to the arrival place and pick up  $c_3$  on time. Instead time is wasted while the crane waits for  $c_2$  to arrive after placing  $c_1$ . This situation can only be avoided if looking ahead when positioning containers so that the next container(s)' arrival time(s) can be taken into consideration when selecting a position. The principle of the exemplified situation is illustrated in figure 5.5.

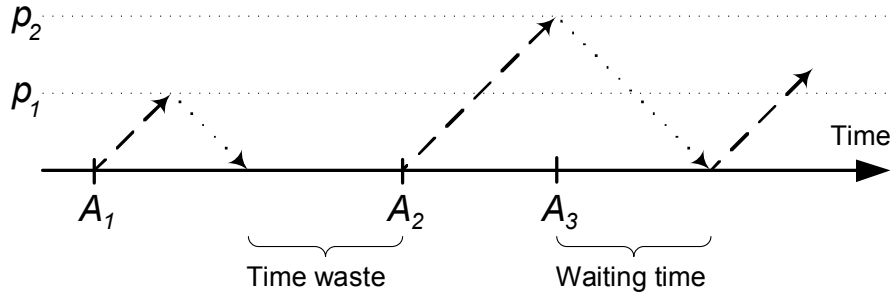


Figure 5.5: Overview of the possible time waste when choosing the closest position, potentially leading to waiting time at the arrival place for subsequent containers.

In all small-scale problem instances - due to no coincident departure times - no moves are performed by criterion 1, in general, the number of positionings chosen by criterion 0 exceeds the number of criterion 2 selections and few moves are performed by criterion

3, the stair instances being exceptions as the arrival and departure time structure makes positionings on top of containers with earlier departure times unavoidable. There is no effect of running the algorithm, not allowing rejections, as no containers are rejected by the basic heuristic. Furthermore, there is no room for improvement by the shifting routine in the scaled instances which is due to the density of arrival and departure times, leaving no time intervals between moves large enough to shift late outgoing moves.

All solutions are as close to optimum as possible, given the initial choices of positions for arrival containers. As discussed above, the only way to avoid or minimize poor initial positionings is to reconsider the position selection strategy, possibly performing a series of runs with different distance criteria for choosing positions and keeping the solution with the least waiting time and delay. Concluding, the heuristic algorithm reaches the best possible solutions with the implemented routines and functions and run times are very short.

As well as in the small-scale test phase, all medium-scale class problems are solved in at most 0.1 seconds by the basic heuristic algorithm and, in general, even less time is spent on running the version without rejections and the shifting routine. The number of full moves  $\#M$  equals the lower bound  $LB$  in the majority of the instances, again with the stair types as obvious exceptions and  $QUAL \geq 85.00\%$  for all other instances.

The criteria statistics resemble the pattern from the small-scale solutions with criterion 0 and 2 dominating, few moves by criterion 3 and none by criterion 1. For obvious reasons there is a clear connection between the number of positionings by criterion 3 and the solution quality, measured in  $GAP$  and  $QUAL$ . Some unavoidable waiting and/or delay occurs in several of the tight and a few of the scaled instances, **10S.s** being the only case where the shifting routine improves the initial solution, eliminating a delay of 14 by increasing the total transportation time by 2.00. Contrary to the small-scale results, rejections occur in a single instance in the medium-scale class. In the **6M6.t** instance, one container is rejected, causing it to be early by 14 and resulting in a gap of -16.67 % and a quality over 100 %. When not allowing rejections, one container is slightly delayed,  $GAP = 0.00\%$ , and  $QUAL = 100.00\%$ . In all remaining medium-scale instances, no rejections occur when running the basic heuristic algorithm.

For the scaled instances' part, the effect of the shifting routine stands out in tables 5.22 and 5.25, included to illustrate exactly this property. The Tuning 0 runs for the 8 containers subclass scaled instances, presented in table 5.22, provide two examples of solutions with room for improvement, one of them being repaired by the shifting routine. In the **8S.s** instance, one container is late but no time intervals with the crane being idle is long enough for the delayed outgoing move to be shifted towards its scheduled time. On the other hand, the solution of the **8M6.s** instance where two containers are delayed is improved by the shifting routine, first mending the outgoing move of the most delayed container by shifting it to the scheduled departure time and, next, transferring the other delayed container's outgoing move, resulting in no delays after running the shifting routine. In table 5.25, the **10S.s** instance provides another example of an initial solution, including a total delay of 14, repaired by the improvement routine, finding room for one late outgoing move to be shifted to the container's scheduled departure

time.

Figures 5.6 and 5.7 show the improvement of the 8M6.s solution made by the shifting routine, illustrated by an extract of the time horizon where the changes are applied. In the initial heuristic solution, container 4, depicted by a red line, is delayed by 8 and container 2, depicted by a green line, is delayed by 2. However, there is crane idle time between 75 and 79, possibly allowing the late outgoing moves to be shifted towards the containers' departure times. The shifting routine repairs the solution by treating the most delayed container first, checking if its outgoing move can be performed earlier, so that the departure time can be met without delaying other containers. This is possible if performing the outgoing move of container 4 immediately after reshuffling the light green container and postponing the reshuffle of the green container 2, which is allowed as they do not share positions. Subsequently, the second-most delayed container, number 2, can be moved unhindered to the departure place in time, thereby eliminating the last delay in the initial heuristic solution.

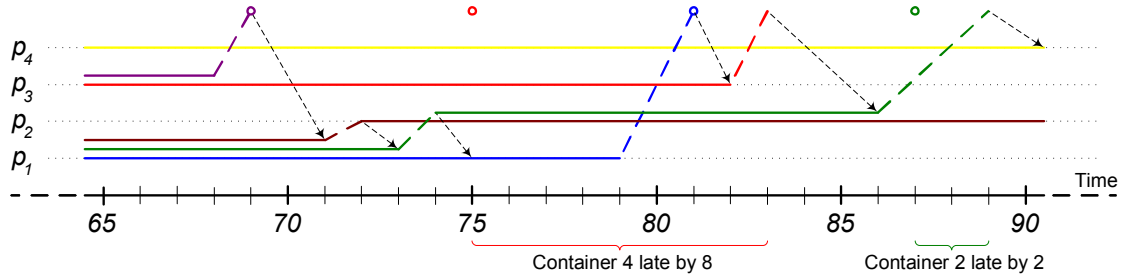


Figure 5.6: Overview of the initial heuristic solution where the red and green containers are late by 8 and 2 respectively. Each of the solid colored lines represents a container, stored at a position for a certain time interval, the sloped dashed lines correspond to crane moves with container of the the given color, and the dark dashed arrows represent empty crane moves. The shifting routine seeks to eliminate the delays by checking if one or both move(s) can be performed closer to the scheduled departure time(s).

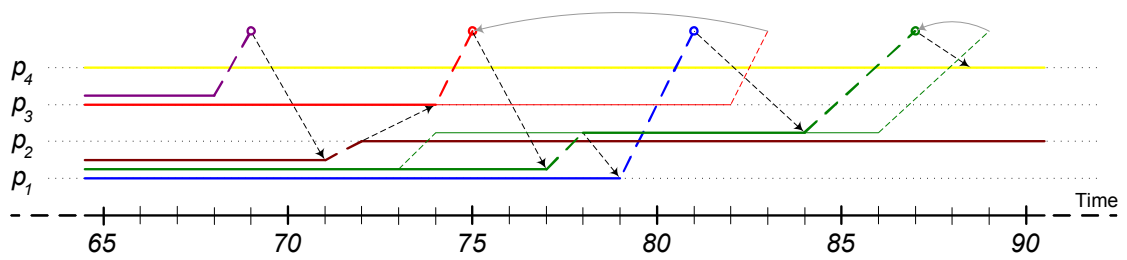


Figure 5.7: Overview of the improved solution after running the shifting routine, changed parts of the initial solution faintly indicated by thin solid and dashed lines. The late outgoing move of the red container is shifted to meet the scheduled departure time, causing the reshuffle of the green container to be postponed and releasing time for shifting its late outgoing move to its departure time.

As well as for the small-scale class problems, the heuristic performs very well in general, producing optimal, or close to optimal, solutions and solving the instances in insignificantly short run times.

### 5.2.3 Large-scale class results

This section presents the results from the large-scale test phase, consisting of real-life problem instances with 856, 1356, 1860, and 2245 containers respectively. Tables 5.28 - 5.39 display the results for the 1 week subclass, differing by the search range  $\rho = 25\% P$ ,  $33\% P$ ,  $50\% P$ ,  $67\% P$ ,  $75\% P$ , and  $100\% P$ , as described in section 4.2.2 on page 100. Furthermore, two search strategies, a combined forward and backward search and a strict forward search, are tested, resulting in 72 runs in each subclass. Equally, tables 5.40 - 5.51 contain the results for the 2 weeks subclass with different values of the search range  $\rho$  and the two different search strategies, and tables 5.52 - 5.63 and 5.64 - 5.75 present the results for the 3 and 4 weeks subclasses respectively.

Each table marked with (f) if concerning a strict forward search, represents six variants of the same problem instance, corresponding to different sizes and proportions of the block, explained in section 4.2.2 as well. The instance name states the number of containers  $C$ , the number of positions  $P$ , and the ratio of the number of rows and bays  $R/B$  in percent. If solving the instance by applying a strict forward search strategy an f is appended to the instance name. As an example, 856.98.50.f denotes a 1 week instance with 856 containers, 98 positions in the block, 7 rows and 14 bays, as  $R = 50\% B$ , and a strict forward search strategy. As explained in the section introduction, each row starting with the problem instance name states the results for the basic heuristic algorithm, rows starting with -no R represent results for the algorithm when not allowing rejections, and rows starting with -shift display the outcome of attempts to repair the initial solutions by the shifting routine. Keys to the columns are identical to previous result tables and can be reread in table 5.13 on page 132.

Solutions to the large-scale class problems number a total of 288 as four subclasses, six different values of  $\rho$ , six different block dimensions, and two search strategies are investigated. Furthermore, each solution takes up a lot of space for which reason they are enclosed on a CD-rom to be found in appendix I.3.



| 1 week<br>$\rho = 25\%$ | Solution |           |              |                  |              | Criteria |     |    |     |     | Performance |       |       |     |
|-------------------------|----------|-----------|--------------|------------------|--------------|----------|-----|----|-----|-----|-------------|-------|-------|-----|
|                         | T        | #M        | Wait         | Early            | Late         | R        | 0   | 1  | 2   | 3   | LB          | GAP   | QUAL  | CPU |
| 856.98.50               | 1,344.04 | 2021/2017 | 241/323.84   | 117/789,177.04   | 220/349.48   | 3/0.35   | 100 | 30 | 743 | 295 | 1712        | 18.05 | 81.95 | 1.3 |
| - no R                  | 1,347.96 | 2027/2023 | 243/330.76   | 114/2,831.04     | 224/359.28   | 0/0.00   | 100 | 31 | 745 | 295 | 1712        | 18.40 | 81.60 | 1.5 |
| - shift                 | 1,347.96 | -         | -            | 114/2,831.04     | 224/359.28   | -        | -   | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.126.40              | 1,617.16 | 1977/1975 | 269/503.56   | 124/789,765.76   | 266/596.68   | 3/0.35   | 96  | 36 | 731 | 261 | 1712        | 15.48 | 84.52 | 1.2 |
| - no R                  | 1,622.60 | 1983/1981 | 269/511.28   | 121/3,419.76     | 270/615.12   | 0/0.00   | 96  | 38 | 732 | 261 | 1712        | 15.83 | 84.17 | 1.5 |
| - shift                 | 1,622.60 | -         | -            | 121/3,419.76     | 270/615.12   | -        | -   | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.161.30              | 1,932.56 | 1931/1929 | 290/839.32   | 104/1,049,615.72 | 352/1,234.12 | 4/0.47   | 99  | 35 | 717 | 228 | 1712        | 12.79 | 87.21 | 1.2 |
| - no R                  | 1,948.08 | 1958/1955 | 291/831.64   | 101/2,929.36     | 354/1,223.96 | 0/0.00   | 100 | 37 | 727 | 238 | 1712        | 14.37 | 85.63 | 1.4 |
| - shift                 | 1,948.08 | -         | -            | 101/2,929.36     | 354/1,223.96 | -        | -   | -  | -   | -   | -           | -     | -     | 0.1 |
| 856.128.50              | 1,541.56 | 2027/2025 | 240/381.08   | 124/788,057.84   | 244/464.96   | 3/0.35   | 97  | 33 | 741 | 303 | 1712        | 18.40 | 81.60 | 1.2 |
| - no R                  | 1,546.44 | 2034/2030 | 240/390.36   | 121/3,331.84     | 249/478.84   | 0/0.00   | 97  | 33 | 744 | 304 | 1712        | 18.81 | 81.19 | 1.3 |
| - shift                 | 1,546.44 | -         | -            | 121/3,331.84     | 249/478.84   | -        | -   | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.160.40              | 1,770.12 | 1944/1942 | 250/531.88   | 125/788,297.88   | 310/766.92   | 3/0.35   | 95  | 38 | 710 | 248 | 1712        | 13.55 | 86.45 | 1.2 |
| - no R                  | 1,779.96 | 1963/1960 | 250/531.00   | 119/3,085.72     | 310/753.36   | 0/0.00   | 95  | 43 | 711 | 258 | 1712        | 14.66 | 85.34 | 1.2 |
| - shift                 | 1,779.96 | -         | -            | 119/3,085.72     | 310/753.36   | -        | -   | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.216.30              | 2,232.56 | 1865/1863 | 314/1,438.00 | 96/1,573,048.44  | 390/1,665.08 | 6/0.70   | 94  | 26 | 711 | 184 | 1712        | 8.94  | 91.06 | 1.2 |
| - no R                  | 2,247.52 | 1877/1874 | 316/1,546.60 | 90/2,019.76      | 400/1,767.88 | 0/0.00   | 94  | 27 | 716 | 184 | 1712        | 9.64  | 90.36 | 1.2 |
| - shift                 | 2,247.52 | -         | -            | 90/2,019.76      | 400/1,767.88 | -        | -   | -  | -   | -   | -           | -     | -     | 0.1 |

Table 5.28: 1 week instances solved with a combined forward and backward search and  $\rho = 25\%$   $P$ .

| 1 week (f)    |          |           | Solution     |                  |              |        | Criteria |    |     |     | Performance |       |       |     |
|---------------|----------|-----------|--------------|------------------|--------------|--------|----------|----|-----|-----|-------------|-------|-------|-----|
| $\rho = 25\%$ | T        | #M        | Wait         | Early            | Late         | R      | 0        | 1  | 2   | 3   | LB          | GAP   | QUAL  | CPU |
| 856.98.50.f   | 1,335.72 | 1986/1980 | 225/283.48   | 111/3,098.24     | 220/305.08   | 0/0.00 | 98       | 28 | 737 | 267 | 1712        | 16.00 | 84.00 | 1.3 |
| - no R        | 1,341.88 | 2003/1996 | 227/281.28   | 115/3,380.08     | 219/314.84   | 0/0.00 | 97       | 28 | 737 | 285 | 1712        | 17.00 | 83.00 | 1.2 |
| - shift       | 1,341.88 | -         | -            | 115/3,380.08     | 219/314.84   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.126.40.f  | 1,617.64 | 1978/1976 | 266/494.44   | 124/789,761.72   | 266/594.00   | 3/0.35 | 96       | 37 | 731 | 261 | 1712        | 15.54 | 84.46 | 1.2 |
| - no R        | 1,623.08 | 1984/1982 | 266/502.16   | 121/3,415.72     | 270/612.44   | 0/0.00 | 96       | 39 | 732 | 261 | 1712        | 15.89 | 84.11 | 1.2 |
| - shift       | 1,623.08 | -         | -            | 121/3,415.72     | 270/612.44   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.161.30.f  | 1,932.56 | 1931/1929 | 290/839.32   | 104/1,049,615.72 | 352/1,235.40 | 4/0.47 | 99       | 35 | 717 | 228 | 1712        | 12.79 | 87.21 | 1.2 |
| - no R        | 1,948.08 | 1958/1955 | 291/831.64   | 101/2,929.36     | 354/1,223.96 | 0/0.00 | 100      | 37 | 727 | 238 | 1712        | 14.37 | 85.63 | 1.3 |
| - shift       | 1,948.08 | -         | -            | 101/2,929.36     | 354/1,223.96 | -      | -        | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.128.50.f  | 1,541.08 | 2027/2025 | 238/375.84   | 124/788,062.92   | 242/458.16   | 3/0.35 | 97       | 33 | 741 | 303 | 1712        | 18.40 | 81.60 | 1.3 |
| - no R        | 1,545.96 | 2034/2030 | 238/385.12   | 121/3,336.92     | 247/472.04   | 0/0.00 | 97       | 33 | 744 | 304 | 1712        | 18.81 | 81.19 | 1.3 |
| - shift       | 1,545.96 | -         | -            | 121/3,336.92     | 247/472.04   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.160.40.f  | 1,770.12 | 1944/1942 | 250/531.88   | 125/788,297.88   | 310/766.92   | 3/0.35 | 95       | 38 | 710 | 248 | 1712        | 13.55 | 86.45 | 1.2 |
| - no R        | 1,779.96 | 1963/1960 | 250/531.00   | 119/3,085.72     | 310/753.36   | 0/0.00 | 95       | 43 | 711 | 258 | 1712        | 14.66 | 85.34 | 1.2 |
| - shift       | 1,779.96 | -         | -            | 119/3,085.72     | 310/753.36   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0 |
| 856.216.30.f  | 2,232.56 | 1865/1863 | 314/1,438.00 | 96/1,573,048.44  | 390/1,665.08 | 6/0.70 | 94       | 26 | 711 | 184 | 1712        | 8.94  | 91.06 | 1.2 |
| - no R        | 2,247.52 | 1877/1874 | 316/1,546.60 | 90/2,019.76      | 400/1,767.88 | 0/0.00 | 94       | 27 | 716 | 184 | 1712        | 9.64  | 90.36 | 1.3 |
| - shift       | 2,247.52 | -         | -            | 90/2,019.76      | 400/1,767.88 | -      | -        | -  | -   | -   | -           | -     | -     | 0.1 |

Table 5.29: 1 week instances solved with a strict forward search and  $\rho = 25\%$   $P$ .

| 1 week<br>$\rho = 33\%$ | Solution |           |              |                  |              | Criteria |    |    |     | Performance |      |       |       |     |
|-------------------------|----------|-----------|--------------|------------------|--------------|----------|----|----|-----|-------------|------|-------|-------|-----|
|                         | T        | #M        | Wait         | Early            | Late         | R        | 0  | 1  | 2   | 3           | LB   | GAP   | QUAL  | CPU |
| 856.98.50               | 1,336.52 | 1982/1980 | 230/286.72   | 115/265,290.72   | 213/330.92   | 1/0.12   | 95 | 30 | 736 | 266         | 1712 | 15.77 | 84.23 | 1.3 |
| - no R                  | 1,338.04 | 1984/1982 | 230/289.32   | 114/3,172.72     | 214/334.60   | 0/0.00   | 95 | 30 | 737 | 266         | 1712 | 15.89 | 84.11 | 1.3 |
| - shift                 | 1,338.04 | -         | -            | 114/3,172.72     | 214/334.60   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.126.40              | 1,591.40 | 1916/1913 | 242/324.12   | 110/526,781.28   | 255/403.56   | 2/0.23   | 95 | 30 | 720 | 217         | 1712 | 11.92 | 88.08 | 1.2 |
| - no R                  | 1,594.44 | 1920/1917 | 244/328.72   | 108/2,553.28     | 258/410.96   | 0/0.00   | 95 | 31 | 721 | 217         | 1712 | 12.15 | 87.85 | 1.3 |
| - shift                 | 1,594.44 | -         | -            | 108/2,553.28     | 258/410.96   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.161.30              | 1,923.76 | 1844/1842 | 283/736.28   | 110/787,543.28   | 340/894.88   | 3/0.35   | 99 | 35 | 687 | 170         | 1712 | 7.71  | 92.29 | 1.2 |
| - no R                  | 1,930.24 | 1850/1848 | 284/746.76   | 107/2,818.40     | 343/916.00   | 0/0.00   | 98 | 36 | 690 | 170         | 1712 | 8.06  | 91.94 | 1.2 |
| - shift                 | 1,930.24 | -         | -            | 107/2,818.40     | 343/916.00   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.128.50              | 1,531.64 | 1994/1992 | 242/421.72   | 126/2,918.76     | 252/447.84   | 0/0.00   | 95 | 38 | 729 | 276         | 1712 | 16.47 | 83.53 | 1.2 |
| - no R                  | 1,531.64 | 1994/1992 | 242/421.72   | 126/2,918.76     | 252/447.84   | 0/0.00   | 95 | 38 | 729 | 276         | 1712 | 16.47 | 83.53 | 1.2 |
| - shift                 | 1,531.64 | -         | -            | 126/2,918.76     | 252/447.84   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.160.40              | 1,754.20 | 1866/1862 | 253/499.96   | 116/787,513.60   | 304/629.40   | 3/0.35   | 99 | 38 | 686 | 190         | 1712 | 9.00  | 91.00 | 1.2 |
| - no R                  | 1,760.36 | 1872/1868 | 254/511.72   | 113/2,790.88     | 309/650.48   | 0/0.00   | 99 | 39 | 688 | 190         | 1712 | 9.35  | 90.65 | 1.3 |
| - shift                 | 1,760.36 | -         | -            | 113/2,790.88     | 309/650.48   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.216.30              | 2,220.08 | 1818/1816 | 313/1,200.56 | 111/1,311,550.28 | 376/1,702.28 | 5/0.58   | 98 | 42 | 691 | 136         | 1712 | 6.19  | 93.81 | 1.2 |
| - no R                  | 2,233.12 | 1828/1825 | 315/1,272.96 | 106/2,612.24     | 383/1,766.64 | 0/0.00   | 98 | 42 | 696 | 136         | 1712 | 6.78  | 93.22 | 1.2 |
| - shift                 | 2,233.12 | -         | -            | 106/2,612.24     | 383/1,766.64 | -        | -  | -  | -   | -           | -    | -     | -     | 0.1 |

Table 5.30: 1 week instances solved with a combined forward and backward search and  $\rho = 33\%$   $P$ .

| 1 week (f)    |          | Solution  |              |                  |              |        | Criteria |    |     |     | Performance |       |       |      |     |
|---------------|----------|-----------|--------------|------------------|--------------|--------|----------|----|-----|-----|-------------|-------|-------|------|-----|
|               |          | T         | #M           | Wait             | Early        | Late   | R        | 0  | 1   | 2   | 3           | LB    | GAP   | QUAL | CPU |
| $\rho = 33\%$ |          |           |              |                  |              |        |          |    |     |     |             |       |       |      |     |
| 856.98.50.f   | 1,335.08 | 1975/1973 | 223/259.20   | 110/265,262.80   | 224/335.24   | 1/0.12 | 99       | 32 | 736 | 253 | 1712        | 15.36 | 84.64 | 1.2  |     |
| - no R        | 1,336.44 | 1977/1975 | 224/261.20   | 109/3,144.80     | 225/337.48   | 0/0.00 | 99       | 32 | 737 | 253 | 1712        | 15.48 | 84.52 | 1.2  |     |
| - shift       | 1,336.44 | -         | -            | 109/3,144.80     | 225/337.48   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0  |     |
| 856.126.40.f  | 1,591.72 | 1916/1913 | 242/322.92   | 110/526,778.32   | 254/400.44   | 2/0.23 | 95       | 30 | 720 | 217 | 1712        | 11.92 | 88.08 | 1.2  |     |
| - no R        | 1,594.76 | 1920/1917 | 244/327.52   | 108/2,550.32     | 257/407.84   | 0/0.00 | 95       | 31 | 721 | 217 | 1712        | 12.15 | 87.85 | 1.3  |     |
| - shift       | 1,594.76 | -         | -            | 108/2,550.32     | 257/407.84   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0  |     |
| 856.161.30.f  | 1,923.76 | 1844/1842 | 283/736.28   | 110/787,543.28   | 340/894.88   | 3/0.35 | 99       | 35 | 687 | 170 | 1712        | 7.71  | 92.29 | 1.2  |     |
| - no R        | 1,930.24 | 1850/1848 | 284/746.76   | 107/2,818.40     | 343/916.00   | 0/0.00 | 98       | 36 | 690 | 170 | 1712        | 8.06  | 91.94 | 1.2  |     |
| - shift       | 1,930.24 | -         | -            | 107/2,818.40     | 343/916.00   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0  |     |
| 856.128.50.f  | 1,532.04 | 1994/1992 | 241/414.64   | 126/2,919.08     | 252/447.20   | 0/0.00 | 95       | 38 | 729 | 276 | 1712        | 16.47 | 83.53 | 1.2  |     |
| - no R        | 1,532.04 | 1994/1992 | 241/414.64   | 126/2,919.08     | 252/447.20   | 0/0.00 | 95       | 38 | 729 | 276 | 1712        | 16.47 | 83.53 | 1.3  |     |
| - shift       | 1,532.04 | -         | -            | 126/2,919.08     | 252/447.20   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0  |     |
| 856.160.40.f  | 1,754.20 | 1866/1862 | 253/499.96   | 116/787,513.60   | 304/629.40   | 3/0.35 | 99       | 38 | 686 | 190 | 1712        | 9.00  | 91.00 | 1.2  |     |
| - no R        | 1,760.36 | 1872/1868 | 254/511.72   | 113/2,790.88     | 309/650.48   | 0/0.00 | 99       | 39 | 688 | 190 | 1712        | 9.35  | 90.65 | 1.4  |     |
| - shift       | 1,760.36 | -         | -            | 113/2,790.88     | 309/650.48   | -      | -        | -  | -   | -   | -           | -     | -     | 0.0  |     |
| 856.216.30.f  | 2,220.08 | 1818/1816 | 313/1,200.56 | 111/1,311,550.28 | 376/1,702.28 | 5/0.58 | 98       | 42 | 691 | 136 | 1712        | 6.19  | 93.81 | 1.3  |     |
| - no R        | 2,233.12 | 1828/1825 | 315/1,272.96 | 106/2,612.24     | 383/1,766.64 | 0/0.00 | 98       | 42 | 696 | 136 | 1712        | 6.78  | 93.22 | 1.5  |     |
| - shift       | 2,233.12 | -         | -            | 106/2,612.24     | 383/1,766.64 | -      | -        | -  | -   | -   | -           | -     | -     | 0.3  |     |

Table 5.31: 1 week instances solved with a strict forward search and  $\rho = 33\%$   $P$ .

| 1 week<br>$\rho = 50\%$ | Solution |           |              |                  |              | Criteria |    |    |     | Performance |      |       |       |     |
|-------------------------|----------|-----------|--------------|------------------|--------------|----------|----|----|-----|-------------|------|-------|-------|-----|
|                         | T        | #M        | Wait         | Early            | Late         | R        | 0  | 1  | 2   | 3           | LB   | GAP   | QUAL  | CPU |
| 856.98.50               | 1,325.08 | 1920/1916 | 226/263.64   | 113/264,613.04   | 200/246.80   | 1/0.12   | 96 | 27 | 723 | 219         | 1712 | 12.15 | 87.85 | 1.2 |
| - no R                  | 1,326.76 | 1922/1917 | 227/265.12   | 113/2,496.12     | 201/248.36   | 0/0.00   | 96 | 28 | 723 | 219         | 1712 | 12.27 | 87.73 | 1.5 |
| - shift                 | 1,326.76 | -         | -            | 113/2,496.12     | 201/248.36   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.126.40              | 1,564.04 | 1795/1790 | 241/301.80   | 108/264,462.60   | 264/376.80   | 1/0.12   | 98 | 32 | 686 | 124         | 1712 | 4.85  | 95.15 | 1.2 |
| - no R                  | 1,565.88 | 1797/1792 | 242/305.08   | 107/2,348.60     | 266/380.32   | 0/0.00   | 98 | 33 | 686 | 124         | 1712 | 4.96  | 95.04 | 1.4 |
| - shift                 | 1,565.88 | -         | -            | 107/2,348.60     | 266/380.32   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.161.30              | 1,910.08 | 1796/1793 | 274/608.52   | 100/526,124.80   | 343/825.00   | 2/0.23   | 97 | 40 | 693 | 112         | 1712 | 4.91  | 95.09 | 1.3 |
| - no R                  | 1,915.52 | 1800/1798 | 274/626.68   | 98/1,896.80      | 345/854.68   | 0/0.00   | 97 | 40 | 695 | 112         | 1712 | 5.14  | 94.86 | 1.4 |
| - shift                 | 1,915.52 | -         | -            | 98/1,896.80      | 345/854.68   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.128.50              | 1,472.04 | 1809/1805 | 214/258.80   | 116/2,838.20     | 253/352.56   | 0/0.00   | 97 | 32 | 688 | 136         | 1712 | 5.67  | 94.33 | 1.2 |
| - no R                  | 1,472.04 | 1809/1805 | 214/258.80   | 116/2,838.20     | 253/352.56   | 0/0.00   | 97 | 32 | 688 | 136         | 1712 | 5.67  | 94.33 | 1.3 |
| - shift                 | 1,472.04 | -         | -            | 116/2,838.20     | 253/352.56   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.160.40              | 1,733.72 | 1803/1800 | 240/404.80   | 107/526,140.44   | 303/587.08   | 2/0.23   | 97 | 37 | 694 | 121         | 1712 | 5.32  | 94.68 | 1.3 |
| - no R                  | 1,738.20 | 1807/1804 | 241/414.52   | 105/1,912.44     | 305/598.72   | 0/0.00   | 97 | 37 | 696 | 121         | 1712 | 5.55  | 94.45 | 1.4 |
| - shift                 | 1,738.20 | -         | -            | 105/1,912.44     | 305/598.72   | -        | -  | -  | -   | -           | -    | -     | -     | 0.0 |
| 856.216.30              | 2,211.44 | 1792/1791 | 311/1,272.08 | 104/1,573,451.36 | 386/1,946.40 | 6/0.70   | 96 | 31 | 696 | 119         | 1712 | 4.67  | 95.33 | 1.4 |
| - no R                  | 2,227.28 | 1804/1802 | 315/1,395.84 | 98/2,413.16      | 393/2,027.24 | 0/0.00   | 96 | 32 | 701 | 119         | 1712 | 5.37  | 94.63 | 1.5 |
| - shift                 | 2,227.28 | -         | -            | 98/2,413.16      | 393/2,027.24 | -        | -  | -  | -   | -           | -    | -     | -     | 0.2 |

Table 5.32: 1 week instances solved with a combined forward and backward search and  $\rho = 50\%$   $P$ .

| 1 week (f)    |          | Solution  |              |                  |              |        |    | Criteria |     |     | Performance |      |       |      |     |
|---------------|----------|-----------|--------------|------------------|--------------|--------|----|----------|-----|-----|-------------|------|-------|------|-----|
|               |          | T         | #M           | Wait             | Early        | Late   | R  | 0        | 1   | 2   | 3           | LB   | GAP   | QUAL | CPU |
| $\rho = 50\%$ |          |           |              |                  |              |        |    |          |     |     |             |      |       |      |     |
| 856.98.50.f   | 1,320.12 | 1876/1870 | 220/247.72   | 123/264,948.52   | 200/220.36   | 1/0.12 | 95 | 37       | 697 | 192 | 1712        | 9.58 | 90.42 | 1.2  |     |
| - no R        | 1,321.32 | 1878/1872 | 221/249.40   | 122/2,830.52     | 201/222.36   | 0/0.00 | 95 | 37       | 698 | 192 | 1712        | 9.70 | 90.30 | 1.3  |     |
| - shift       | 1,321.32 | -         | -            | 122/2,830.52     | 201/222.36   | -      | -  | -        | -   | -   | -           | -    | -     | 0.0  |     |
| 856.126.40.f  | 1,564.20 | 1795/1790 | 241/300.44   | 107/264,463.08   | 265/380.60   | 1/0.12 | 98 | 32       | 686 | 124 | 1712        | 4.85 | 95.15 | 1.2  |     |
| - no R        | 1,566.04 | 1797/1792 | 242/303.72   | 106/2,349.08     | 267/384.12   | 0/0.00 | 98 | 33       | 686 | 124 | 1712        | 4.96 | 95.04 | 1.3  |     |
| - shift       | 1,566.04 | -         | -            | 106/2,349.08     | 267/384.12   | -      | -  | -        | -   | -   | -           | -    | -     | 0.0  |     |
| 856.161.30.f  | 1,910.08 | 1796/1793 | 274/608.52   | 100/526,124.80   | 343/825.00   | 2/0.23 | 97 | 40       | 693 | 112 | 1712        | 4.91 | 95.09 | 1.2  |     |
| - no R        | 1,915.52 | 1800/1798 | 274/626.68   | 98/1,896.80      | 345/854.68   | 0/0.00 | 97 | 40       | 695 | 112 | 1712        | 5.14 | 94.86 | 1.3  |     |
| - shift       | 1,915.52 | -         | -            | 98/1,896.80      | 345/854.68   | -      | -  | -        | -   | -   | -           | -    | -     | 0.0  |     |
| 856.128.50.f  | 1,472.04 | 1809/1805 | 214/258.80   | 115/2,834.92     | 253/350.60   | 0/0.00 | 97 | 32       | 688 | 136 | 1712        | 5.67 | 94.33 | 1.2  |     |
| - no R        | 1,472.04 | 1809/1805 | 214/258.80   | 115/2,834.92     | 253/350.60   | 0/0.00 | 97 | 32       | 688 | 136 | 1712        | 5.67 | 94.33 | 1.3  |     |
| - shift       | 1,472.04 | -         | -            | 115/2,834.92     | 253/350.60   | -      | -  | -        | -   | -   | -           | -    | -     | 0.0  |     |
| 856.160.40.f  | 1,733.72 | 1803/1800 | 240/404.80   | 107/526,140.44   | 303/587.08   | 2/0.23 | 97 | 37       | 694 | 121 | 1712        | 5.32 | 94.68 | 1.2  |     |
| - no R        | 1,738.20 | 1807/1804 | 241/414.52   | 105/1,912.44     | 305/598.72   | 0/0.00 | 97 | 37       | 696 | 121 | 1712        | 5.55 | 94.45 | 1.3  |     |
| - shift       | 1,738.20 | -         | -            | 105/1,912.44     | 305/598.72   | -      | -  | -        | -   | -   | -           | -    | -     | 0.0  |     |
| 856.216.30.f  | 2,211.44 | 1792/1791 | 311/1,272.08 | 104/1,573,451.36 | 386/1,946.40 | 6/0.70 | 96 | 31       | 696 | 119 | 1712        | 4.67 | 95.33 | 1.3  |     |
| - no R        | 2,227.28 | 1804/1802 | 315/1,395.84 | 98/2,413.16      | 393/2,027.24 | 0/0.00 | 96 | 32       | 701 | 119 | 1712        | 5.37 | 94.63 | 1.3  |     |
| - shift       | 2,227.28 | -         | -            | 98/2,413.16      | 393/2,027.24 | -      | -  | -        | -   | -   | -           | -    | -     | 0.2  |     |

Table 5.33: 1 week instances solved with a strict forward search and  $\rho = 50\%$   $P$ .

| 1 week<br>$\rho = 67\%$ | Solution |           |              |                 |              | Criteria |    |    |     | Performance |      |      |       |     |
|-------------------------|----------|-----------|--------------|-----------------|--------------|----------|----|----|-----|-------------|------|------|-------|-----|
|                         | T        | #M        | Wait         | Early           | Late         | R        | 0  | 1  | 2   | 3           | LB   | GAP  | QUAL  | CPU |
| 856.98.50               | 1,301.40 | 1815/1812 | 218/225.12   | 102/2,087.40    | 215/255.96   | 0/0.00   | 98 | 27 | 704 | 130         | 1712 | 6.02 | 93.98 | 1.2 |
| - no R                  | 1,301.40 | 1815/1812 | 218/225.12   | 102/2,087.40    | 215/255.96   | 0/0.00   | 98 | 27 | 704 | 130         | 1712 | 6.02 | 93.98 | 1.4 |
| - shift                 | 1,301.40 | -         | -            | 102/2,087.40    | 215/255.96   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.126.40              | 1,570.04 | 1801/1797 | 240/322.40   | 113/2,020.96    | 252/382.08   | 0/0.00   | 97 | 37 | 687 | 124         | 1712 | 5.20 | 94.80 | 1.2 |
| - no R                  | 1,570.04 | 1801/1797 | 240/322.40   | 113/2,020.96    | 252/382.08   | 0/0.00   | 97 | 37 | 687 | 124         | 1712 | 5.20 | 94.80 | 1.4 |
| - shift                 | 1,570.04 | -         | -            | 113/2,020.96    | 252/382.08   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.161.30              | 1,912.00 | 1811/1808 | 271/597.80   | 102/526,266.00  | 343/839.52   | 2/0.23   | 96 | 33 | 698 | 130         | 1712 | 5.78 | 94.22 | 1.3 |
| - no R                  | 1,917.60 | 1815/1813 | 272/613.72   | 100/2,038.00    | 346/865.16   | 0/0.00   | 96 | 33 | 700 | 130         | 1712 | 6.02 | 93.98 | 1.3 |
| - shift                 | 1,917.60 | -         | -            | 100/2,038.00    | 346/865.16   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.128.50              | 1,461.16 | 1800/1796 | 214/243.00   | 105/2,048.72    | 243/315.08   | 0/0.00   | 97 | 33 | 697 | 117         | 1712 | 5.14 | 94.86 | 1.2 |
| - no R                  | 1,461.16 | 1800/1796 | 214/243.00   | 105/2,048.72    | 243/315.08   | 0/0.00   | 97 | 33 | 697 | 117         | 1712 | 5.14 | 94.86 | 1.3 |
| - shift                 | 1,461.16 | -         | -            | 105/2,048.72    | 243/315.08   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.160.40              | 1,738.12 | 1792/1789 | 241/456.84   | 112/526,272.96  | 289/558.12   | 2/0.23   | 95 | 41 | 687 | 115         | 1712 | 4.67 | 95.33 | 1.3 |
| - no R                  | 1,743.40 | 1796/1794 | 242/466.64   | 110/2,044.96    | 292/572.08   | 0/0.00   | 95 | 41 | 689 | 115         | 1712 | 4.91 | 95.09 | 1.3 |
| - shift                 | 1,743.40 | -         | -            | 110/2,044.96    | 292/572.08   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.216.30              | 2,208.80 | 1787/1786 | 310/1,225.40 | 97/1,573,228.28 | 391/1,838.12 | 6/0.70   | 97 | 37 | 694 | 109         | 1712 | 4.38 | 95.62 | 1.3 |
| - no R                  | 2,224.88 | 1799/1797 | 313/1,354.04 | 89/2,190.44     | 400/1,932.04 | 0/0.00   | 97 | 38 | 699 | 109         | 1712 | 5.08 | 94.92 | 1.3 |
| - shift                 | 2,224.88 | -         | -            | 89/2,190.44     | 400/1,932.04 | -        | -  | -  | -   | -           | -    | -    | -     | 0.1 |

Table 5.34: 1 week instances solved with a combined forward and backward search and  $\rho = 67\%$   $P$ .

| 1 week (f)   | Solution |           |              |                 |              |        | Criteria |    |     |     | Performance |      |       |     |
|--------------|----------|-----------|--------------|-----------------|--------------|--------|----------|----|-----|-----|-------------|------|-------|-----|
|              | T        | #M        | Wait         | Early           | Late         | R      | 0        | 1  | 2   | 3   | LB          | GAP  | QUAL  | CPU |
| 856.98.50.f  | 1,294.84 | 1809/1805 | 213/229.00   | 106/2,266.32    | 205/207.04   | 0/0.00 | 97       | 28 | 698 | 130 | 1712        | 5.67 | 94.33 | 1.8 |
| - no R       | 1,294.84 | 1809/1805 | 213/229.00   | 106/2,266.32    | 205/207.04   | 0/0.00 | 97       | 28 | 698 | 130 | 1712        | 5.67 | 94.33 | 1.3 |
| - shift      | 1,294.84 | -         | -            | 106/2,266.32    | 205/207.04   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.126.40.f | 1,569.88 | 1801/1796 | 238/308.96   | 112/2,005.04    | 252/372.44   | 0/0.00 | 97       | 37 | 687 | 124 | 1712        | 5.20 | 94.80 | 1.4 |
| - no R       | 1,569.88 | 1801/1796 | 238/308.96   | 112/2,005.04    | 252/372.44   | 0/0.00 | 97       | 37 | 687 | 124 | 1712        | 5.20 | 94.80 | 1.3 |
| - shift      | 1,569.88 | -         | -            | 112/2,005.04    | 252/372.44   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.161.30.f | 1,912.00 | 1811/1808 | 271/597.80   | 102/526,266.00  | 343/839.52   | 2/0.23 | 96       | 33 | 698 | 130 | 1712        | 5.78 | 94.22 | 1.3 |
| - no R       | 1,917.60 | 1815/1813 | 272/613.72   | 100/2,038.00    | 346/865.16   | 0/0.00 | 96       | 33 | 700 | 130 | 1712        | 6.02 | 93.98 | 1.3 |
| - shift      | 1,917.60 | -         | -            | 100/2,038.00    | 346/865.16   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.128.50.f | 1,461.00 | 1800/1796 | 212/238.40   | 103/2,047.60    | 243/309.88   | 0/0.00 | 97       | 33 | 697 | 117 | 1712        | 5.14 | 94.86 | 1.2 |
| - no R       | 1,461.00 | 1800/1796 | 212/238.40   | 103/2,047.60    | 243/309.88   | 0/0.00 | 97       | 33 | 697 | 117 | 1712        | 5.14 | 94.86 | 1.4 |
| - shift      | 1,461.00 | -         | -            | 103/2,047.60    | 243/309.88   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.160.40.f | 1,738.12 | 1792/1789 | 241/456.84   | 112/526,272.96  | 289/558.12   | 2/0.23 | 95       | 41 | 687 | 115 | 1712        | 4.67 | 95.33 | 1.3 |
| - no R       | 1,743.40 | 1796/1794 | 242/466.64   | 110/2,044.96    | 292/572.08   | 0/0.00 | 95       | 41 | 689 | 115 | 1712        | 4.91 | 95.09 | 1.4 |
| - shift      | 1,743.40 | -         | -            | 110/2,044.96    | 292/572.08   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.216.30.f | 2,208.80 | 1787/1786 | 310/1,225.40 | 97/1,573,228.28 | 391/1,838.12 | 6/0.70 | 97       | 37 | 694 | 109 | 1712        | 4.38 | 95.62 | 1.3 |
| - no R       | 2,224.88 | 1799/1797 | 313/1,354.04 | 89/2,190.44     | 400/1,932.04 | 0/0.00 | 97       | 38 | 699 | 109 | 1712        | 5.08 | 94.92 | 1.5 |
| - shift      | 2,224.88 | -         | -            | 89/2,190.44     | 400/1,932.04 | -      | -        | -  | -   | -   | -           | -    | -     | 0.2 |

Table 5.35: 1 week instances solved with a strict forward search and  $\rho = 67\%$   $P$ .



| 1 week<br>$\rho = 75\%$ | Solution |           |              |                 |              | Criteria |    |    |     | Performance |      |      |       |     |
|-------------------------|----------|-----------|--------------|-----------------|--------------|----------|----|----|-----|-------------|------|------|-------|-----|
|                         | T        | #M        | Wait         | Early           | Late         | R        | 0  | 1  | 2   | 3           | LB   | GAP  | QUAL  | CPU |
| 856.98.50               | 1,297.88 | 1790/1786 | 216/230.80   | 105/264,200.48  | 202/219.48   | 1/0.12   | 98 | 34 | 689 | 114         | 1712 | 4.56 | 95.44 | 1.3 |
| - no R                  | 1,299.80 | 1792/1789 | 217/234.28   | 104/2,086.48    | 204/223.32   | 0/0.00   | 98 | 34 | 690 | 114         | 1712 | 4.67 | 95.33 | 1.4 |
| - shift                 | 1,299.80 | -         | -            | 104/2,086.48    | 204/223.32   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.126.40              | 1,572.20 | 1802/1798 | 238/311.24   | 103/263,882.72  | 268/402.96   | 1/0.12   | 96 | 33 | 700 | 118         | 1712 | 5.26 | 94.74 | 1.3 |
| - no R                  | 1,574.04 | 1804/1800 | 239/313.80   | 102/1,768.72    | 270/406.00   | 0/0.00   | 96 | 34 | 700 | 118         | 1712 | 5.37 | 94.63 | 1.3 |
| - shift                 | 1,574.04 | -         | -            | 102/1,768.72    | 270/406.00   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.161.30              | 1,912.00 | 1811/1808 | 271/597.80   | 102/526,266.00  | 343/839.52   | 2/0.23   | 96 | 33 | 698 | 130         | 1712 | 5.78 | 94.22 | 1.3 |
| - no R                  | 1,917.60 | 1815/1813 | 272/613.72   | 100/2,038.00    | 346/865.16   | 0/0.00   | 96 | 33 | 700 | 130         | 1712 | 6.02 | 93.98 | 1.3 |
| - shift                 | 1,917.60 | -         | -            | 100/2,038.00    | 346/865.16   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.128.50              | 1,466.04 | 1800/1796 | 223/294.44   | 114/2,092.28    | 232/339.00   | 0/0.00   | 96 | 37 | 690 | 121         | 1712 | 5.14 | 94.86 | 1.3 |
| - no R                  | 1,466.04 | 1800/1796 | 223/294.44   | 114/2,092.28    | 232/339.00   | 0/0.00   | 96 | 37 | 690 | 121         | 1712 | 5.14 | 94.86 | 1.3 |
| - shift                 | 1,466.04 | -         | -            | 114/2,092.28    | 232/339.00   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.160.40              | 1,738.12 | 1792/1789 | 241/456.84   | 112/526,272.96  | 289/558.12   | 2/0.23   | 95 | 41 | 687 | 115         | 1712 | 4.67 | 95.33 | 1.4 |
| - no R                  | 1,743.40 | 1796/1794 | 242/466.64   | 110/2,044.96    | 292/572.08   | 0/0.00   | 95 | 41 | 689 | 115         | 1712 | 4.91 | 95.09 | 1.4 |
| - shift                 | 1,743.40 | -         | -            | 110/2,044.96    | 292/572.08   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.216.30              | 2,208.80 | 1787/1786 | 310/1,225.40 | 97/1,573,228.28 | 391/1,838.12 | 6/0.70   | 97 | 37 | 694 | 109         | 1712 | 4.38 | 95.62 | 1.3 |
| - no R                  | 2,224.88 | 1799/1797 | 313/1,354.04 | 89/2,190.44     | 400/1,932.04 | 0/0.00   | 97 | 38 | 699 | 109         | 1712 | 5.08 | 94.92 | 1.7 |
| - shift                 | 2,224.88 | -         | -            | 89/2,190.44     | 400/1,932.04 | -        | -  | -  | -   | -           | -    | -    | -     | 0.2 |

Table 5.36: 1 week instances solved with a combined forward and backward search and  $\rho = 75\%$   $P$ .

| 1 week (f)   | Solution |           |              |                 |              |        | Criteria |    |     |     | Performance |      |       |     |
|--------------|----------|-----------|--------------|-----------------|--------------|--------|----------|----|-----|-----|-------------|------|-------|-----|
|              | T        | #M        | Wait         | Early           | Late         | R      | 0        | 1  | 2   | 3   | LB          | GAP  | QUAL  | CPU |
| 856.98.50.f  | 1,298.68 | 1796/1792 | 217/236.16   | 109/2,676.32    | 203/214.36   | 0/0.00 | 97       | 32 | 685 | 126 | 1712        | 4.91 | 95.09 | 1.2 |
| - no R       | 1,298.68 | 1796/1792 | 217/236.16   | 109/2,676.32    | 203/214.36   | 0/0.00 | 97       | 32 | 685 | 126 | 1712        | 4.91 | 95.09 | 1.3 |
| - shift      | 1,298.68 | -         | -            | 109/2,676.32    | 203/214.36   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.126.40.f | 1,572.04 | 1802/1797 | 236/298.12   | 101/263,874.04  | 268/395.80   | 1/0.12 | 96       | 33 | 700 | 118 | 1712        | 5.26 | 94.74 | 1.2 |
| - no R       | 1,573.88 | 1804/1799 | 237/300.68   | 100/1,760.04    | 270/398.84   | 0/0.00 | 96       | 34 | 700 | 118 | 1712        | 5.37 | 94.63 | 1.4 |
| - shift      | 1,573.88 | -         | -            | 100/1,760.04    | 270/398.84   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.161.30.f | 1,912.00 | 1811/1808 | 271/597.80   | 102/526,266.00  | 343/839.52   | 2/0.23 | 96       | 33 | 698 | 130 | 1712        | 5.78 | 94.22 | 1.3 |
| - no R       | 1,917.60 | 1815/1813 | 272/613.72   | 100/2,038.00    | 346/865.16   | 0/0.00 | 96       | 33 | 700 | 130 | 1712        | 6.02 | 93.98 | 1.3 |
| - shift      | 1,917.60 | -         | -            | 100/2,038.00    | 346/865.16   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.128.50.f | 1,465.72 | 1800/1796 | 222/286.04   | 112/2,080.56    | 232/323.96   | 0/0.00 | 96       | 37 | 690 | 121 | 1712        | 5.14 | 94.86 | 1.3 |
| - no R       | 1,465.72 | 1800/1796 | 222/286.04   | 112/2,080.56    | 232/323.96   | 0/0.00 | 96       | 37 | 690 | 121 | 1712        | 5.14 | 94.86 | 1.4 |
| - shift      | 1,465.72 | -         | -            | 112/2,080.56    | 232/323.96   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.160.40.f | 1,739.80 | 1801/1798 | 238/443.08   | 111/526,393.40  | 291/556.80   | 2/0.23 | 95       | 39 | 695 | 118 | 1712        | 5.20 | 94.80 | 1.3 |
| - no R       | 1,745.08 | 1805/1803 | 240/452.64   | 109/2,165.40    | 294/570.44   | 0/0.00 | 95       | 39 | 697 | 118 | 1712        | 5.43 | 94.57 | 1.4 |
| - shift      | 1,745.08 | -         | -            | 109/2,165.40    | 294/570.44   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.216.30.f | 2,208.80 | 1787/1786 | 310/1,225.40 | 97/1,573,228.28 | 391/1,838.12 | 6/0.70 | 97       | 37 | 694 | 109 | 1712        | 4.38 | 95.62 | 1.3 |
| - no R       | 2,224.88 | 1799/1797 | 313/1,354.04 | 89/2,190.44     | 400/1,932.04 | 0/0.00 | 97       | 38 | 699 | 109 | 1712        | 5.08 | 94.92 | 1.5 |
| - shift      | 2,224.88 | -         | -            | 89/2,190.44     | 400/1,932.04 | -      | -        | -  | -   | -   | -           | -    | -     | 0.1 |

Table 5.37: 1 week instances solved with a strict forward search and  $\rho = 75\%$   $P$ .

| 1 week<br>$\rho = 100\%$ | Solution |           |              |                 |              | Criteria |    |    |     | Performance |      |      |       |     |
|--------------------------|----------|-----------|--------------|-----------------|--------------|----------|----|----|-----|-------------|------|------|-------|-----|
|                          | T        | #M        | Wait         | Early           | Late         | R        | 0  | 1  | 2   | 3           | LB   | GAP  | QUAL  | CPU |
| 856.98.50                | 1,300.04 | 1798/1793 | 224/245.48   | 103/1,870.76    | 204/217.80   | 0/0.00   | 95 | 37 | 693 | 117         | 1712 | 5.02 | 94.98 | 1.2 |
| - no R                   | 1,300.04 | 1798/1793 | 224/245.48   | 103/1,870.76    | 204/217.80   | 0/0.00   | 95 | 37 | 693 | 117         | 1712 | 5.02 | 94.98 | 1.4 |
| - shift                  | 1,300.04 | -         | -            | 103/1,870.76    | 204/217.80   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.126.40               | 1,574.52 | 1798/1793 | 245/342.08   | 109/1,925.96    | 258/377.44   | 0/0.00   | 96 | 39 | 695 | 112         | 1712 | 5.02 | 94.98 | 1.3 |
| - no R                   | 1,574.52 | 1798/1793 | 245/342.08   | 109/1,925.96    | 258/377.44   | 0/0.00   | 96 | 39 | 695 | 112         | 1712 | 5.02 | 94.98 | 1.4 |
| - shift                  | 1,574.52 | -         | -            | 109/1,925.96    | 258/377.44   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.161.30               | 1,911.76 | 1791/1787 | 271/604.28   | 105/526,542.72  | 327/737.80   | 2/0.23   | 95 | 36 | 691 | 115         | 1712 | 4.61 | 95.39 | 1.3 |
| - no R                   | 1,916.72 | 1795/1792 | 272/616.68   | 103/2,314.72    | 330/760.24   | 0/0.00   | 95 | 36 | 693 | 115         | 1712 | 4.85 | 95.15 | 1.3 |
| - shift                  | 1,916.72 | -         | -            | 103/2,314.72    | 330/760.24   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.128.50               | 1,466.92 | 1799/1795 | 221/291.16   | 107/2,061.56    | 241/357.16   | 0/0.00   | 97 | 36 | 694 | 116         | 1712 | 5.08 | 94.92 | 1.3 |
| - no R                   | 1,466.92 | 1799/1795 | 221/291.16   | 107/2,061.56    | 241/357.16   | 0/0.00   | 97 | 36 | 694 | 116         | 1712 | 5.08 | 94.92 | 1.4 |
| - shift                  | 1,466.92 | -         | -            | 107/2,061.56    | 241/357.16   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.160.40               | 1,739.80 | 1801/1798 | 238/443.08   | 111/526,393.40  | 291/556.80   | 2/0.23   | 95 | 39 | 695 | 118         | 1712 | 5.20 | 94.80 | 1.3 |
| - no R                   | 1,745.08 | 1805/1803 | 240/452.64   | 109/2,165.40    | 294/570.44   | 0/0.00   | 95 | 39 | 697 | 118         | 1712 | 5.43 | 94.57 | 1.3 |
| - shift                  | 1,745.08 | -         | -            | 109/2,165.40    | 294/570.44   | -        | -  | -  | -   | -           | -    | -    | -     | 0.0 |
| 856.216.30               | 2,208.80 | 1787/1786 | 310/1,225.40 | 97/1,573,228.28 | 391/1,838.12 | 6/0.70   | 97 | 37 | 694 | 109         | 1712 | 4.38 | 95.62 | 1.4 |
| - no R                   | 2,224.88 | 1799/1797 | 313/1,354.04 | 89/2,190.44     | 400/1,932.04 | 0/0.00   | 97 | 38 | 699 | 109         | 1712 | 5.08 | 94.92 | 1.5 |
| - shift                  | 2,224.88 | -         | -            | 89/2,190.44     | 400/1,932.04 | -        | -  | -  | -   | -           | -    | -    | -     | 0.2 |

Table 5.38: 1 week instances solved with a combined forward and backward search and  $\rho = 100\%$   $P$ .

| 1 week (f)     | Solution |           |              |                 |              |        | Criteria |    |     |     | Performance |      |       |     |
|----------------|----------|-----------|--------------|-----------------|--------------|--------|----------|----|-----|-----|-------------|------|-------|-----|
|                | T        | #M        | Wait         | Early           | Late         | R      | 0        | 1  | 2   | 3   | LB          | GAP  | QUAL  | CPU |
| $\rho = 100\%$ |          |           |              |                 |              |        |          |    |     |     |             |      |       |     |
| 856.98.50.f    | 1,302.20 | 1793/1789 | 223/228.64   | 103/1,895.28    | 199/199.04   | 0/0.00 | 95       | 38 | 693 | 111 | 1712        | 4.73 | 95.27 | 1.3 |
| - no R         | 1,302.20 | 1793/1789 | 223/228.64   | 103/1,895.28    | 199/199.04   | 0/0.00 | 95       | 38 | 693 | 111 | 1712        | 4.73 | 95.27 | 1.3 |
| - shift        | 1,302.20 | -         | -            | 103/1,895.28    | 199/199.04   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.126.40.f   | 1,574.60 | 1798/1793 | 243/332.40   | 106/1,913.28    | 260/376.28   | 0/0.00 | 96       | 39 | 695 | 112 | 1712        | 5.02 | 94.98 | 1.3 |
| - no R         | 1,574.60 | 1798/1793 | 243/332.40   | 106/1,913.28    | 260/376.28   | 0/0.00 | 96       | 39 | 695 | 112 | 1712        | 5.02 | 94.98 | 1.3 |
| - shift        | 1,574.60 | -         | -            | 106/1,913.28    | 260/376.28   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.161.30.f   | 1,911.76 | 1791/1787 | 271/604.28   | 105/526,542.72  | 327/737.80   | 2/0.23 | 95       | 36 | 691 | 115 | 1712        | 4.61 | 95.39 | 1.3 |
| - no R         | 1,916.72 | 1795/1792 | 272/616.68   | 103/2,314.72    | 330/760.24   | 0/0.00 | 95       | 36 | 693 | 115 | 1712        | 4.85 | 95.15 | 1.3 |
| - shift        | 1,916.72 | -         | -            | 103/2,314.72    | 330/760.24   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.128.50.f   | 1,466.52 | 1799/1795 | 220/282.04   | 105/2,049.68    | 240/335.56   | 0/0.00 | 97       | 36 | 694 | 116 | 1712        | 5.08 | 94.92 | 1.3 |
| - no R         | 1,466.52 | 1799/1795 | 220/282.04   | 105/2,049.68    | 240/335.56   | 0/0.00 | 97       | 36 | 694 | 116 | 1712        | 5.08 | 94.92 | 1.3 |
| - shift        | 1,466.52 | -         | -            | 105/2,049.68    | 240/335.56   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.160.40.f   | 1,739.80 | 1801/1798 | 238/443.08   | 111/526,393.40  | 291/556.80   | 2/0.23 | 95       | 39 | 695 | 118 | 1712        | 5.20 | 94.80 | 1.4 |
| - no R         | 1,745.08 | 1805/1803 | 240/452.64   | 109/2,165.40    | 294/570.44   | 0/0.00 | 95       | 39 | 697 | 118 | 1712        | 5.43 | 94.57 | 1.4 |
| - shift        | 1,745.08 | -         | -            | 109/2,165.40    | 294/570.44   | -      | -        | -  | -   | -   | -           | -    | -     | 0.0 |
| 856.216.30.f   | 2,208.80 | 1787/1786 | 310/1,225.40 | 97/1,573,228.28 | 391/1,838.12 | 6/0.70 | 97       | 37 | 694 | 109 | 1712        | 4.38 | 95.62 | 1.4 |
| - no R         | 2,224.88 | 1799/1797 | 313/1,354.04 | 89/2,190.44     | 400/1,932.04 | 0/0.00 | 97       | 38 | 699 | 109 | 1712        | 5.08 | 94.92 | 1.4 |
| - shift        | 2,224.88 | -         | -            | 89/2,190.44     | 400/1,932.04 | -      | -        | -  | -   | -   | -           | -    | -     | 0.1 |

Table 5.39: 1 week instances solved with a strict forward search and  $\rho = 100\%$   $P$ .

| 2 weeks       |  | Solution |           |              |                  |              |         | Criteria |    |      |     | Performance |       |       |     |
|---------------|--|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|-------|-------|-----|
| $\rho = 25\%$ |  | T        | #M        | Wait         | Early            | Late         | R       | 0        | 1  | 2    | 3   | LB          | GAP   | QUAL  | CPU |
| 1356.98.50    |  | 2,158.52 | 3266/3260 | 408/605.68   | 200/1,331,596.56 | 375/650.44   | 5/0.37  | 140      | 49 | 1203 | 523 | 2712        | 20.43 | 79.57 | 3.5 |
| - no R        |  | 2,166.84 | 3291/3283 | 409/615.36   | 189/5,420.28     | 385/651.44   | 0/0.00  | 141      | 49 | 1216 | 529 | 2712        | 21.35 | 78.65 | 3.1 |
| - shift       |  | 2,166.84 | -         | -            | 189/5,420.28     | 385/651.44   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1356.126.40   |  | 2,570.12 | 3158/3155 | 468/874.64   | 192/1,331,923.36 | 480/1,123.32 | 5/0.37  | 131      | 51 | 1182 | 443 | 2712        | 16.45 | 83.55 | 3.3 |
| - no R        |  | 2,578.52 | 3172/3164 | 468/884.12   | 187/5,948.28     | 486/1,148.00 | 0/0.00  | 131      | 53 | 1186 | 446 | 2712        | 16.96 | 83.04 | 3.0 |
| - shift       |  | 2,578.52 | -         | -            | 187/5,948.28     | 486/1,148.00 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1356.161.30   |  | 3,089.28 | 3087/3085 | 512/1,638.60 | 171/1,860,394.04 | 592/2,186.00 | 7/0.52  | 133      | 60 | 1150 | 395 | 2712        | 13.83 | 86.17 | 3.2 |
| - no R        |  | 3,110.64 | 3124/3117 | 514/1,649.40 | 168/5,405.36     | 588/2,083.32 | 0/0.00  | 135      | 56 | 1169 | 408 | 2712        | 15.19 | 84.81 | 3.1 |
| - shift       |  | 3,110.64 | -         | -            | 168/5,405.36     | 588/2,083.32 | -       | -        | -  | -    | -   | -           | -     | -     | 0.1 |
| 1356.128.50   |  | 2,444.92 | 3258/3255 | 417/700.68   | 203/1,329,885.04 | 423/922.16   | 5/0.37  | 138      | 48 | 1193 | 528 | 2712        | 20.13 | 79.87 | 3.1 |
| - no R        |  | 2,452.44 | 3270/3262 | 417/709.96   | 198/5,530.36     | 431/937.72   | 0/0.00  | 138      | 48 | 1198 | 530 | 2712        | 20.58 | 79.42 | 3.0 |
| - shift       |  | 2,452.44 | -         | -            | 198/5,530.36     | 431/937.72   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1356.160.40   |  | 2,830.60 | 3139/3136 | 464/1,361.12 | 188/1,330,272.00 | 544/1,545.72 | 5/0.37  | 136      | 45 | 1166 | 441 | 2712        | 15.74 | 84.26 | 3.3 |
| - no R        |  | 2,837.40 | 3167/3159 | 473/1,239.96 | 186/5,661.24     | 540/1,410.96 | 0/0.00  | 136      | 55 | 1164 | 456 | 2712        | 16.78 | 83.22 | 3.1 |
| - shift       |  | 2,837.40 | -         | -            | 186/5,661.24     | 540/1,410.96 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1356.216.30   |  | 3,540.80 | 2973/2971 | 560/2,717.72 | 157/2,926,288.20 | 667/3,197.76 | 11/0.81 | 127      | 44 | 1146 | 311 | 2712        | 9.62  | 90.38 | 3.3 |
| - no R        |  | 3,578.88 | 3003/2997 | 566/2,845.40 | 147/3,495.56     | 677/3,348.52 | 0/0.00  | 127      | 48 | 1154 | 318 | 2712        | 10.73 | 89.27 | 3.1 |
| - shift       |  | 3,578.88 | -         | -            | 147/3,495.56     | 677/3,348.52 | -       | -        | -  | -    | -   | -           | -     | -     | 0.2 |

Table 5.40: 2 weeks instances solved with a combined forward and backward search and  $\rho = 25\%$   $P$ .

| 2 weeks (f)   |          | Solution  |              |                  |              |         | Criteria |    |      |     | Performance |       |       |      |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|-------|-------|------|
|               |          | T         | #M           | Wait             | Early        | Late    | R        | 0  | 1    | 2   | 3           | LB    | GAP   | QUAL |
| 1356.98.50.f  | 2,156.36 | 3258/3250 | 407/583.16   | 196/5,45,669.60  | 370/610.12   | 2/0.15  | 134      | 43 | 1208 | 519 | 2712        | 20.13 | 79.87 | 3.0  |
| - no R        | 2,159.00 | 3267/3256 | 403/582.32   | 191/5,738.36     | 379/617.20   | 0/0.00  | 132      | 42 | 1214 | 523 | 2712        | 20.46 | 79.54 | 3.0  |
| - shift       | 2,159.00 | -         | -            | 191/5,738.36     | 379/617.20   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.126.40.f | 2,570.60 | 3158/3155 | 464/864.64   | 192/1,331,919.08 | 481/1,164.72 | 5/0.37  | 131      | 52 | 1181 | 443 | 2712        | 16.45 | 83.55 | 3.0  |
| - no R        | 2,579.00 | 3172/3164 | 464/874.12   | 187/5,944.00     | 487/1,189.40 | 0/0.00  | 131      | 54 | 1185 | 446 | 2712        | 16.96 | 83.04 | 3.1  |
| - shift       | 2,579.00 | -         | -            | 187/5,944.00     | 487/1,189.40 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.161.30.f | 3,089.28 | 3087/3085 | 512/1,638.60 | 171/1,860,394.04 | 592/2,187.28 | 7/0.52  | 133      | 60 | 1150 | 395 | 2712        | 13.83 | 86.17 | 3.0  |
| - no R        | 3,110.40 | 3124/3117 | 513/1,645.00 | 168/5,405.36     | 589/2,092.12 | 0/0.00  | 135      | 56 | 1169 | 408 | 2712        | 15.19 | 84.81 | 3.0  |
| - shift       | 3,110.40 | -         | -            | 168/5,405.36     | 589/2,092.12 | -       | -        | -  | -    | -   | -           | -     | -     | 0.1  |
| 1356.128.50.f | 2,444.44 | 3258/3255 | 415/695.44   | 203/1,329,890.12 | 421/912.20   | 5/0.37  | 138      | 48 | 1193 | 528 | 2712        | 20.13 | 79.87 | 3.0  |
| - no R        | 2,451.96 | 3270/3262 | 415/704.72   | 198/5,535.44     | 429/927.76   | 0/0.00  | 138      | 48 | 1198 | 530 | 2712        | 20.58 | 79.42 | 3.0  |
| - shift       | 2,451.96 | -         | -            | 198/5,535.44     | 429/927.76   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.160.40.f | 2,830.60 | 3139/3136 | 464/1,361.12 | 188/1,330,272.00 | 544/1,545.72 | 5/0.37  | 136      | 45 | 1166 | 441 | 2712        | 15.74 | 84.26 | 3.1  |
| - no R        | 2,837.40 | 3167/3159 | 473/1,239.96 | 186/5,661.24     | 540/1,410.96 | 0/0.00  | 136      | 55 | 1164 | 456 | 2712        | 16.78 | 83.22 | 3.0  |
| - shift       | 2,837.40 | -         | -            | 186/5,661.24     | 540/1,410.96 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.216.30.f | 3,540.80 | 2973/2971 | 560/2,717.72 | 157/2,926,288.20 | 667/3,197.76 | 11/0.81 | 127      | 44 | 1146 | 311 | 2712        | 9.62  | 90.38 | 3.1  |
| - no R        | 3,578.88 | 3003/2997 | 566/2,845.40 | 147/3,495.56     | 677/3,348.52 | 0/0.00  | 127      | 48 | 1154 | 318 | 2712        | 10.73 | 89.27 | 3.0  |
| - shift       | 3,578.88 | -         | -            | 147/3,495.56     | 677/3,348.52 | -       | -        | -  | -    | -   | -           | -     | -     | 0.2  |

Table 5.41: 2 weeks instances solved with a strict forward search and  $\rho = 25\%$   $P$ .

| 2 weeks<br>$\rho = 33\%$ |          | Solution  |              |                  |              |        |     | Criteria |      |     |      | Performance |       |      |     |
|--------------------------|----------|-----------|--------------|------------------|--------------|--------|-----|----------|------|-----|------|-------------|-------|------|-----|
|                          |          | T         | #M           | Wait             | Early        | Late   | R   | 0        | 1    | 2   | 3    | LB          | GAP   | QUAL | CPU |
| 1356.98.50               | 2,133.16 | 3176/3173 | 394/503.56   | 183/807,051.20   | 364/593.56   | 3/0.22 | 129 | 50       | 1194 | 450 | 2712 | 17.11       | 82.89 | 3.0  |     |
| - no R                   | 2,136.68 | 3182/3177 | 394/506.16   | 180/5,304.52     | 367/598.68   | 0/0.00 | 129 | 50       | 1197 | 450 | 2712 | 17.33       | 82.67 | 3.1  |     |
| - shift                  | 2,136.68 | -         | -            | 180/5,304.52     | 367/598.68   | -      | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 1356.126.40              | 2,551.64 | 3070/3066 | 431/850.00   | 188/1,068,769.04 | 445/853.40   | 4/0.29 | 128 | 45       | 1157 | 388 | 2712 | 13.20       | 86.80 | 3.1  |     |
| - no R                   | 2,557.64 | 3078/3072 | 433/854.60   | 184/4,912.12     | 450/862.64   | 0/0.00 | 128 | 46       | 1160 | 388 | 2712 | 13.50       | 86.50 | 3.2  |     |
| - shift                  | 2,557.64 | -         | -            | 184/4,912.12     | 450/862.64   | -      | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 1356.161.30              | 3,053.68 | 2927/2925 | 502/1,446.36 | 169/1,597,799.60 | 586/1,773.04 | 6/0.44 | 132 | 54       | 1108 | 283 | 2712 | 7.93        | 92.07 | 3.1  |     |
| - no R                   | 3,065.92 | 2939/2934 | 503/1,457.80 | 163/4,262.76     | 592/1,797.60 | 0/0.00 | 131 | 55       | 1114 | 283 | 2712 | 8.37        | 91.63 | 3.0  |     |
| - shift                  | 3,065.92 | -         | -            | 163/4,262.76     | 592/1,797.60 | -      | -   | -        | -    | -   | -    | -           | -     | 0.1  |     |
| 1356.128.50              | 2,414.60 | 3139/3136 | 420/708.68   | 198/545,047.52   | 426/750.72   | 2/0.15 | 133 | 53       | 1162 | 437 | 2712 | 15.74       | 84.26 | 3.1  |     |
| - no R                   | 2,417.64 | 3143/3138 | 420/708.68   | 196/5,421.52     | 428/752.40   | 0/0.00 | 133 | 53       | 1164 | 437 | 2712 | 15.89       | 84.11 | 3.0  |     |
| - shift                  | 2,417.64 | -         | -            | 196/5,421.52     | 428/752.40   | -      | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 1356.160.40              | 2,783.88 | 2970/2965 | 450/1,163.24 | 170/1,328,185.00 | 527/1,427.08 | 5/0.37 | 130 | 58       | 1131 | 300 | 2712 | 9.51        | 90.49 | 3.1  |     |
| - no R                   | 2,793.96 | 2980/2973 | 451/1,175.00 | 165/3,833.28     | 534/1,450.24 | 0/0.00 | 130 | 59       | 1135 | 300 | 2712 | 9.88        | 90.12 | 3.0  |     |
| - shift                  | 2,793.96 | -         | -            | 165/3,833.28     | 534/1,450.24 | -      | -   | -        | -    | -   | -    | -           | -     | 0.1  |     |
| 1356.216.30              | 3,515.04 | 2857/2854 | 554/2,389.72 | 174/1,852,436.32 | 646/3,046.64 | 7/0.52 | 130 | 66       | 1102 | 210 | 2712 | 5.35        | 94.65 | 3.1  |     |
| - no R                   | 3,532.88 | 2871/2865 | 556/2,462.12 | 167/3,872.28     | 655/3,113.56 | 0/0.00 | 130 | 66       | 1109 | 210 | 2712 | 5.86        | 94.14 | 3.0  |     |
| - shift                  | 3,532.88 | -         | -            | 167/3,872.28     | 655/3,113.56 | -      | -   | -        | -    | -   | -    | -           | -     | 0.2  |     |

Table 5.42: 2 weeks instances solved with a combined forward and backward search and  $\rho = 33\% P$ .

| 2 weeks (f)   |          | Solution  |              |                  |              |        | Criteria |    |      |     | Performance |       |       |      |     |
|---------------|----------|-----------|--------------|------------------|--------------|--------|----------|----|------|-----|-------------|-------|-------|------|-----|
|               |          | T         | #M           | Wait             | Early        | Late   | R        | 0  | 1    | 2   | 3           | LB    | GAP   | QUAL | CPU |
| $\rho = 33\%$ |          |           |              |                  |              |        |          |    |      |     |             |       |       |      |     |
| 1356.98.50.f  | 2,135.00 | 3143/3141 | 385/476.08   | 192/1,076,696.84 | 353/502.20   | 4/0.29 | 135      | 50 | 1179 | 427 | 2712        | 15.89 | 84.11 | 3.8  |     |
| - no R        | 2,142.52 | 3155/3150 | 387/479.00   | 182/5,713.52     | 367/547.04   | 0/0.00 | 137      | 48 | 1184 | 430 | 2712        | 16.33 | 83.67 | 3.0  |     |
| - shift       | 2,142.52 | -         | -            | 182/5,713.52     | 367/547.04   | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |     |
| 1356.126.40.f | 2,551.56 | 3070/3066 | 431/848.80   | 188/1,068,755.92 | 444/840.44   | 4/0.29 | 128      | 45 | 1157 | 388 | 2712        | 13.20 | 86.80 | 3.1  |     |
| - no R        | 2,557.56 | 3078/3072 | 433/853.40   | 184/4,899.00     | 449/849.68   | 0/0.00 | 128      | 46 | 1160 | 388 | 2712        | 13.50 | 86.50 | 3.1  |     |
| - shift       | 2,557.56 | -         | -            | 184/4,899.00     | 449/849.68   | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |     |
| 1356.161.30.f | 3,053.68 | 2927/2925 | 502/1,446.36 | 169/1,597,799.60 | 586/1,773.04 | 6/0.44 | 132      | 54 | 1108 | 283 | 2712        | 7.93  | 92.07 | 3.2  |     |
| - no R        | 3,065.92 | 2939/2934 | 503/1,457.80 | 163/4,262.76     | 592/1,797.60 | 0/0.00 | 131      | 55 | 1114 | 283 | 2712        | 8.37  | 91.63 | 3.1  |     |
| - shift       | 3,065.92 | -         | -            | 163/4,262.76     | 592/1,797.60 | -      | -        | -  | -    | -   | -           | -     | -     | 0.1  |     |
| 1356.128.50.f | 2,415.00 | 3139/3136 | 419/701.60   | 198/545,047.84   | 426/750.08   | 2/0.15 | 133      | 53 | 1162 | 437 | 2712        | 15.74 | 84.26 | 3.1  |     |
| - no R        | 2,418.04 | 3143/3138 | 419/701.60   | 196/5,421.84     | 428/751.76   | 0/0.00 | 133      | 53 | 1164 | 437 | 2712        | 15.89 | 84.11 | 3.1  |     |
| - shift       | 2,418.04 | -         | -            | 196/5,421.84     | 428/751.76   | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |     |
| 1356.160.40.f | 2,783.88 | 2970/2965 | 450/1,163.24 | 170/1,328,185.00 | 527/1,427.08 | 5/0.37 | 130      | 58 | 1131 | 300 | 2712        | 9.51  | 90.49 | 3.1  |     |
| - no R        | 2,793.96 | 2980/2973 | 451/1,175.00 | 165/3,833.28     | 534/1,450.24 | 0/0.00 | 130      | 59 | 1135 | 300 | 2712        | 9.88  | 90.12 | 3.0  |     |
| - shift       | 2,793.96 | -         | -            | 165/3,833.28     | 534/1,450.24 | -      | -        | -  | -    | -   | -           | -     | -     | 0.1  |     |
| 1356.216.30.f | 3,515.04 | 2857/2854 | 554/2,389.72 | 174/1,852,436.32 | 646/3,046.64 | 7/0.52 | 130      | 66 | 1102 | 210 | 2712        | 5.35  | 94.65 | 3.1  |     |
| - no R        | 3,532.88 | 2871/2865 | 556/2,462.12 | 167/3,872.28     | 655/3,113.56 | 0/0.00 | 130      | 66 | 1109 | 210 | 2712        | 5.86  | 94.14 | 3.1  |     |
| - shift       | 3,532.88 | -         | -            | 167/3,872.28     | 655/3,113.56 | -      | -        | -  | -    | -   | -           | -     | -     | 0.2  |     |

Table 5.43: 2 weeks instances solved with a strict forward search and  $\rho = 33\%$   $P$ .



| 2 weeks<br>$\rho = 50\%$ |          | Solution  |              |                  |              |        | Criteria |    |      |     | Performance |       |       |      |
|--------------------------|----------|-----------|--------------|------------------|--------------|--------|----------|----|------|-----|-------------|-------|-------|------|
|                          |          | T         | #M           | Wait             | Early        | Late   | R        | 0  | 1    | 2   | 3           | LB    | GAP   | QUAL |
| 1356.98.50               | 2,120.92 | 3050/3044 | 379/452.80   | 174/806,086.52   | 365/491.64   | 3/0.22 | 126      | 42 | 1175 | 354 | 2712        | 12.46 | 87.54 | 3.2  |
| - no R                   | 2,125.16 | 3056/3047 | 380/454.28   | 172/4,343.60     | 368/494.80   | 0/0.00 | 126      | 43 | 1177 | 354 | 2712        | 12.68 | 87.32 | 3.0  |
| - shift                  | 2,125.16 | -         | -            | 172/4,343.60     | 368/494.80   | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.126.40              | 2,492.36 | 2837/2831 | 404/548.20   | 180/805,454.20   | 440/589.12   | 3/0.22 | 133      | 53 | 1101 | 197 | 2712        | 4.61  | 95.39 | 3.1  |
| - no R                   | 2,497.40 | 2843/2835 | 405/551.48   | 177/3,711.28     | 444/594.56   | 0/0.00 | 133      | 54 | 1103 | 197 | 2712        | 4.83  | 95.17 | 3.1  |
| - shift                  | 2,497.40 | -         | -            | 177/3,711.28     | 444/594.56   | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.161.30              | 3,024.16 | 2821/2817 | 472/1,102.04 | 165/1,067,184.60 | 583/1,515.76 | 4/0.29 | 131      | 61 | 1102 | 175 | 2712        | 4.02  | 95.98 | 3.1  |
| - no R                   | 3,033.76 | 2829/2824 | 472/1,120.20 | 161/3,327.56     | 587/1,547.68 | 0/0.00 | 131      | 61 | 1106 | 175 | 2712        | 4.31  | 95.69 | 3.1  |
| - shift                  | 3,033.76 | -         | -            | 161/3,327.56     | 587/1,547.68 | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.128.50              | 2,334.60 | 2870/2865 | 366/500.24   | 188/543,811.92   | 431/683.88   | 2/0.15 | 127      | 56 | 1113 | 220 | 2712        | 5.83  | 94.17 | 3.1  |
| - no R                   | 2,337.64 | 2874/2867 | 366/500.24   | 186/4,183.00     | 433/685.56   | 0/0.00 | 127      | 56 | 1115 | 220 | 2712        | 5.97  | 94.03 | 3.0  |
| - shift                  | 2,337.64 | -         | -            | 186/4,183.00     | 433/685.56   | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.160.40              | 2,745.08 | 2846/2842 | 423/778.56   | 188/1,067,040.68 | 507/1,028.56 | 4/0.29 | 129      | 58 | 1109 | 198 | 2712        | 4.94  | 95.06 | 3.1  |
| - no R                   | 2,753.00 | 2854/2848 | 424/788.28   | 184/3,183.76     | 511/1,042.04 | 0/0.00 | 129      | 58 | 1113 | 198 | 2712        | 5.24  | 94.76 | 3.1  |
| - shift                  | 2,753.00 | -         | -            | 184/3,183.76     | 511/1,042.04 | -      | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1356.216.30              | 3,501.92 | 2817/2816 | 551/2,550.20 | 164/2,383,387.20 | 646/3,372.96 | 9/0.66 | 126      | 46 | 1119 | 179 | 2712        | 3.87  | 96.13 | 3.1  |
| - no R                   | 3,524.72 | 2835/2830 | 555/2,675.08 | 155/3,536.80     | 656/3,458.44 | 0/0.00 | 126      | 47 | 1127 | 179 | 2712        | 4.54  | 95.46 | 3.2  |
| - shift                  | 3,524.72 | -         | -            | 155/3,536.80     | 656/3,458.44 | -      | -        | -  | -    | -   | -           | -     | -     | 0.2  |

Table 5.44: 2 weeks instances solved with a combined forward and backward search and  $\rho = 50\%$   $P$ .

| 2 weeks (f)   |          | Solution  |              |                  |              |        | Criteria |     |    |      | Performance |      |       |       |     |
|---------------|----------|-----------|--------------|------------------|--------------|--------|----------|-----|----|------|-------------|------|-------|-------|-----|
|               |          | T         | #M           | Wait             | Early        | Late   | R        | 0   | 1  | 2    | 3           | LB   | GAP   | QUAL  | CPU |
| $\rho = 50\%$ |          |           |              |                  |              |        |          |     |    |      |             |      |       |       |     |
| 1356.98.50.f  | 2,114.92 | 3028/3020 | 379/439.60   | 209/807,034.80   | 335/401.48   | 3/0.22 | 3/0.22   | 125 | 58 | 1134 | 358         | 2712 | 11.65 | 88.35 | 3.1 |
| - no R        | 2,118.84 | 3034/3024 | 380/441.28   | 206/5,290.80     | 338/405.00   | 0/0.00 | 0/0.00   | 125 | 58 | 1137 | 358         | 2712 | 11.87 | 88.13 | 3.1 |
| - shift       | 2,118.84 | -         | -            | 206/5,290.80     | 338/405.00   | -      | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 1356.126.40.f | 2,492.60 | 2837/2831 | 404/544.84   | 179/805,453.24   | 441/594.36   | 3/0.22 | 3/0.22   | 133 | 53 | 1101 | 197         | 2712 | 4.61  | 95.39 | 3.2 |
| - no R        | 2,497.64 | 2843/2835 | 405/548.12   | 176/3,710.32     | 445/599.80   | 0/0.00 | 0/0.00   | 133 | 54 | 1103 | 197         | 2712 | 4.83  | 95.17 | 3.1 |
| - shift       | 2,497.64 | -         | -            | 176/3,710.32     | 445/599.80   | -      | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 1356.161.30.f | 3,024.16 | 2821/2817 | 472/1,102.04 | 165/1,067,184.60 | 583/1,515.76 | 4/0.29 | 4/0.29   | 131 | 61 | 1102 | 175         | 2712 | 4.02  | 95.98 | 3.1 |
| - no R        | 3,033.76 | 2829/2824 | 472/1,120.20 | 161/3,327.56     | 587/1,547.68 | 0/0.00 | 0/0.00   | 131 | 61 | 1106 | 175         | 2712 | 4.31  | 95.69 | 3.1 |
| - shift       | 3,033.76 | -         | -            | 161/3,327.56     | 587/1,547.68 | -      | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 1356.128.50.f | 2,334.44 | 2870/2865 | 366/499.68   | 187/543,796.64   | 432/677.80   | 2/0.15 | 2/0.15   | 127 | 56 | 1113 | 220         | 2712 | 5.83  | 94.17 | 3.1 |
| - no R        | 2,337.48 | 2874/2867 | 366/499.68   | 185/4,167.72     | 434/679.48   | 0/0.00 | 0/0.00   | 127 | 56 | 1115 | 220         | 2712 | 5.97  | 94.03 | 3.0 |
| - shift       | 2,337.48 | -         | -            | 185/4,167.72     | 434/679.48   | -      | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 1356.160.40.f | 2,745.08 | 2846/2842 | 423/778.56   | 188/1,067,040.68 | 507/1,028.56 | 4/0.29 | 4/0.29   | 129 | 58 | 1109 | 198         | 2712 | 4.94  | 95.06 | 3.1 |
| - no R        | 2,753.00 | 2854/2848 | 424/788.28   | 184/3,183.76     | 511/1,042.04 | 0/0.00 | 0/0.00   | 129 | 58 | 1113 | 198         | 2712 | 5.24  | 94.76 | 3.1 |
| - shift       | 2,753.00 | -         | -            | 184/3,183.76     | 511/1,042.04 | -      | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 1356.216.30.f | 3,501.92 | 2817/2816 | 551/2,550.20 | 164/2,383,387.20 | 646/3,372.96 | 9/0.66 | 9/0.66   | 126 | 46 | 1119 | 179         | 2712 | 3.87  | 96.13 | 3.1 |
| - no R        | 3,524.72 | 2835/2830 | 555/2,675.08 | 155/3,536.80     | 656/3,458.44 | 0/0.00 | 0/0.00   | 126 | 47 | 1127 | 179         | 2712 | 4.54  | 95.46 | 3.1 |
| - shift       | 3,524.72 | -         | -            | 155/3,536.80     | 656/3,458.44 | -      | -        | -   | -  | -    | -           | -    | -     | -     | 0.3 |

Table 5.45: 2 weeks instances solved with a strict forward search and  $\rho = 50\%$   $P$ .

| 2 weeks<br>$\rho = 67\%$ |          | Solution  |              |                  |              |         | Criteria |    |      |     | Performance |      |       |      |
|--------------------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|------|-------|------|
|                          |          | T         | #M           | Wait             | Early        | Late    | R        | 0  | 1    | 2   | 3           | LB   | GAP   | QUAL |
| 1356.98.50               | 2,066.12 | 2865/2861 | 370/399.92   | 174/543,205.88   | 364/436.28   | 2/0.15  | 134      | 48 | 1121 | 208 | 2712        | 5.64 | 94.36 | 3.1  |
| - no R                   | 2,068.60 | 2869/2863 | 370/399.92   | 172/3,579.88     | 366/437.96   | 0/0.00  | 134      | 48 | 1123 | 208 | 2712        | 5.79 | 94.21 | 3.0  |
| - shift                  | 2,068.60 | -         | -            | 172/3,579.88     | 366/437.96   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0  |
| 1356.126.40              | 2,495.80 | 2837/2832 | 405/585.28   | 181/542,396.72   | 445/662.12   | 2/0.15  | 130      | 56 | 1107 | 190 | 2712        | 4.61 | 95.39 | 3.1  |
| - no R                   | 2,499.08 | 2841/2834 | 405/585.28   | 179/2,770.72     | 447/663.88   | 0/0.00  | 130      | 56 | 1109 | 190 | 2712        | 4.76 | 95.24 | 3.1  |
| - shift                  | 2,499.08 | -         | -            | 179/2,770.72     | 447/663.88   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0  |
| 1356.161.30              | 3,030.64 | 2855/2850 | 481/1,215.48 | 164/1,066,704.96 | 580/1,631.28 | 4/0.29  | 129      | 57 | 1120 | 197 | 2712        | 5.27 | 94.73 | 3.2  |
| - no R                   | 3,040.48 | 2863/2857 | 482/1,231.40 | 160/2,850.96     | 585/1,659.24 | 0/0.00  | 129      | 57 | 1124 | 197 | 2712        | 5.57 | 94.43 | 3.3  |
| - shift                  | 3,040.48 | -         | -            | 160/2,850.96     | 585/1,659.24 | -       | -        | -  | -    | -   | -           | -    | -     | 0.1  |
| 1356.128.50              | 2,309.08 | 2834/2829 | 365/473.96   | 186/542,772.80   | 408/531.16   | 2/0.15  | 128      | 56 | 1108 | 188 | 2712        | 4.50 | 95.50 | 3.1  |
| - no R                   | 2,312.12 | 2838/2831 | 365/473.96   | 184/3,146.80     | 410/532.76   | 0/0.00  | 128      | 56 | 1110 | 188 | 2712        | 4.65 | 95.35 | 3.1  |
| - shift                  | 2,312.12 | -         | -            | 184/3,146.80     | 410/532.76   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0  |
| 1356.160.40              | 2,750.84 | 2823/2819 | 423/855.56   | 171/1,066,908.96 | 511/1,108.00 | 4/0.29  | 129      | 62 | 1103 | 177 | 2712        | 4.09 | 95.91 | 3.2  |
| - no R                   | 2,759.72 | 2831/2826 | 424/865.36   | 167/3,051.96     | 516/1,123.88 | 0/0.00  | 129      | 62 | 1107 | 177 | 2712        | 4.39 | 95.61 | 3.1  |
| - shift                  | 2,759.72 | -         | -            | 167/3,051.96     | 516/1,123.88 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0  |
| 1356.216.30              | 3,490.80 | 2819/2818 | 548/2,386.08 | 156/2,655,212.36 | 653/3,224.52 | 10/0.74 | 126      | 59 | 1116 | 172 | 2712        | 3.95 | 96.05 | 3.2  |
| - no R                   | 3,516.72 | 2839/2834 | 551/2,523.76 | 144/3,167.24     | 666/3,349.72 | 0/0.00  | 126      | 60 | 1125 | 172 | 2712        | 4.68 | 95.32 | 3.2  |
| - shift                  | 3,516.72 | -         | -            | 144/3,167.24     | 666/3,349.72 | -       | -        | -  | -    | -   | -           | -    | -     | 0.3  |

Table 5.46: 2 weeks instances solved with a combined forward and backward search and  $\rho = 67\%$   $P$ .

| 2 weeks (f)   |          |           |              |                  |              |         |     |    |      | Criteria |      |      |       | Performance |     |      |     |
|---------------|----------|-----------|--------------|------------------|--------------|---------|-----|----|------|----------|------|------|-------|-------------|-----|------|-----|
| Solution      |          |           |              |                  |              |         |     |    |      | Criteria |      |      |       | Performance |     |      |     |
|               |          |           |              |                  |              |         |     |    |      | 0        | 1    | 2    | 3     | LB          | GAP | QUAL | CPU |
| $\rho = 67\%$ | T        | #M        | Wait         | Early            | Late         | R       |     |    |      |          |      |      |       |             |     |      |     |
| 1356.98.50.f  | 2,057.08 | 2859/2854 | 360/369.72   | 176/543,404.76   | 337/313.88   | 2/0.15  | 133 | 50 | 1110 | 212      | 2712 | 5.42 | 94.58 | 3.2         |     |      |     |
| - no R        | 2,060.12 | 2863/2856 | 360/369.72   | 174/3,775.92     | 339/315.56   | 0/0.00  | 133 | 50 | 1112 | 212      | 2712 | 5.57 | 94.43 | 3.0         |     |      |     |
| - shift       | 2,060.12 | -         | -            | 174/3,775.92     | 339/315.56   | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |     |      |     |
| 1356.126.40.f | 2,496.44 | 2836/2830 | 402/571.40   | 180/542,381.36   | 445/653.44   | 2/0.15  | 130 | 56 | 1106 | 190      | 2712 | 4.57 | 95.43 | 3.1         |     |      |     |
| - no R        | 2,499.72 | 2840/2832 | 402/571.40   | 178/2,755.36     | 447/655.20   | 0/0.00  | 130 | 56 | 1108 | 190      | 2712 | 4.72 | 95.28 | 3.1         |     |      |     |
| - shift       | 2,499.72 | -         | -            | 178/2,755.36     | 447/655.20   | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |     |      |     |
| 1356.161.30.f | 3,030.64 | 2855/2850 | 481/1,215.48 | 164/1,066,704.96 | 580/1,631.28 | 4/0.29  | 129 | 57 | 1120 | 197      | 2712 | 5.27 | 94.73 | 3.2         |     |      |     |
| - no R        | 3,040.48 | 2863/2857 | 482/1,231.40 | 160/2,850.96     | 585/1,659.24 | 0/0.00  | 129 | 57 | 1124 | 197      | 2712 | 5.57 | 94.43 | 3.1         |     |      |     |
| - shift       | 3,040.48 | -         | -            | 160/2,850.96     | 585/1,659.24 | -       | -   | -  | -    | -        | -    | -    | -     | 0.1         |     |      |     |
| 1356.128.50.f | 2,308.84 | 2833/2828 | 363/466.72   | 182/542,766.04   | 408/526.44   | 2/0.15  | 128 | 56 | 1107 | 188      | 2712 | 4.46 | 95.54 | 3.1         |     |      |     |
| - no R        | 2,311.88 | 2837/2830 | 363/466.72   | 180/3,140.04     | 410/528.04   | 0/0.00  | 128 | 56 | 1109 | 188      | 2712 | 4.61 | 95.39 | 3.1         |     |      |     |
| - shift       | 2,311.88 | -         | -            | 180/3,140.04     | 410/528.04   | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |     |      |     |
| 1356.160.40.f | 2,750.84 | 2823/2819 | 423/855.56   | 171/1,066,908.96 | 511/1,108.00 | 4/0.29  | 129 | 62 | 1103 | 177      | 2712 | 4.09 | 95.91 | 3.2         |     |      |     |
| - no R        | 2,759.72 | 2831/2826 | 424/865.36   | 167/3,051.96     | 516/1,123.88 | 0/0.00  | 129 | 62 | 1107 | 177      | 2712 | 4.39 | 95.61 | 3.1         |     |      |     |
| - shift       | 2,759.72 | -         | -            | 167/3,051.96     | 516/1,123.88 | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |     |      |     |
| 1356.216.30.f | 3,490.80 | 2819/2818 | 548/2,386.08 | 156/2,655,212.36 | 653/3,224.52 | 10/0.74 | 126 | 59 | 1116 | 172      | 2712 | 3.95 | 96.05 | 3.4         |     |      |     |
| - no R        | 3,516.72 | 2839/2834 | 551/2,523.76 | 144/3,167.24     | 666/3,349.72 | 0/0.00  | 126 | 60 | 1125 | 172      | 2712 | 4.68 | 95.32 | 3.2         |     |      |     |
| - shift       | 3,516.72 | -         | -            | 144/3,167.24     | 666/3,349.72 | -       | -   | -  | -    | -        | -    | -    | -     | 0.3         |     |      |     |

Table 5.47: 2 weeks instances solved with a strict forward search and  $\rho = 67\%$   $P$ .

| 2 weeks       |          |           |              |                  |              |         |          |    |      |     |             |      |       |     |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|------|-------|-----|
| $\rho = 75\%$ |          |           |              |                  |              |         |          |    |      |     |             |      |       |     |
| Solution      |          |           |              |                  |              |         | Criteria |    |      |     | Performance |      |       |     |
| T             | #M       | Wait      | Early        | Late             | R            |         | 0        | 1  | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| 1356.98.50    | 2,060.04 | 2826/2821 | 368/406.12   | 176/804,713.88   | 328/330.36   | 3/0.22  | 130      | 58 | 1103 | 182 | 2712        | 4.20 | 95.80 | 3.1 |
| - no R        | 2,063.96 | 2832/2826 | 369/409.60   | 173/2,973.88     | 332/335.56   | 0/0.00  | 130      | 58 | 1106 | 182 | 2712        | 4.42 | 95.58 | 3.1 |
| - shift       | 2,063.96 | -         | -            | 173/2,973.88     | 332/335.56   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.126.40   | 2,497.72 | 2843/2838 | 406/592.04   | 171/804,254.76   | 457/684.92   | 3/0.22  | 129      | 56 | 1124 | 181 | 2712        | 4.83 | 95.17 | 3.1 |
| - no R        | 2,502.52 | 2849/2842 | 407/594.60   | 168/2,514.76     | 461/689.64   | 0/0.00  | 129      | 57 | 1126 | 181 | 2712        | 5.05 | 94.95 | 3.1 |
| - shift       | 2,502.52 | -         | -            | 168/2,514.76     | 461/689.64   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.161.30   | 3,030.64 | 2855/2850 | 481/1,215.48 | 164/1,066,704.96 | 580/1,631.28 | 4/0.29  | 129      | 57 | 1120 | 197 | 2712        | 5.27 | 94.73 | 3.2 |
| - no R        | 3,040.48 | 2863/2857 | 482/1,231.40 | 160/2,850.96     | 585/1,659.24 | 0/0.00  | 129      | 57 | 1124 | 197 | 2712        | 5.57 | 94.43 | 3.1 |
| - shift       | 3,040.48 | -         | -            | 160/2,850.96     | 585/1,659.24 | -       | -        | -  | -    | -   | -           | -    | -     | 0.1 |
| 1356.128.50   | 2,321.08 | 2845/2840 | 387/553.84   | 194/543,285.72   | 393/569.16   | 2/0.15  | 131      | 63 | 1100 | 197 | 2712        | 4.90 | 95.10 | 3.1 |
| - no R        | 2,323.88 | 2849/2842 | 387/553.84   | 192/3,654.20     | 395/570.60   | 0/0.00  | 131      | 63 | 1102 | 197 | 2712        | 5.05 | 94.95 | 3.1 |
| - shift       | 2,323.88 | -         | -            | 192/3,654.20     | 395/570.60   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.160.40   | 2,751.72 | 2823/2819 | 423/855.56   | 171/1,066,908.96 | 511/1,109.92 | 4/0.29  | 129      | 62 | 1103 | 177 | 2712        | 4.09 | 95.91 | 3.1 |
| - no R        | 2,760.60 | 2831/2826 | 424/865.36   | 167/3,051.96     | 516/1,125.80 | 0/0.00  | 129      | 62 | 1107 | 177 | 2712        | 4.39 | 95.61 | 3.1 |
| - shift       | 2,760.60 | -         | -            | 167/3,051.96     | 516/1,125.80 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.216.30   | 3,490.80 | 2819/2818 | 548/2,386.08 | 156/2,655,212.36 | 653/3,224.52 | 10/0.74 | 126      | 59 | 1116 | 172 | 2712        | 3.95 | 96.05 | 3.2 |
| - no R        | 3,516.72 | 2839/2834 | 551/2,523.76 | 144/3,167.24     | 666/3,349.72 | 0/0.00  | 126      | 60 | 1125 | 172 | 2712        | 4.68 | 95.32 | 3.2 |
| - shift       | 3,516.72 | -         | -            | 144/3,167.24     | 666/3,349.72 | -       | -        | -  | -    | -   | -           | -    | -     | 0.3 |

Table 5.48: 2 weeks instances solved with a combined forward and backward search and  $\rho = 75\%$   $P$ .

| 2 weeks (f)   |  | Solution |           |              |                  |              | Criteria |     |    |      | Performance |      |      |       |     |
|---------------|--|----------|-----------|--------------|------------------|--------------|----------|-----|----|------|-------------|------|------|-------|-----|
| $\rho = 75\%$ |  | T        | #M        | Wait         | Early            | Late         | R        | 0   | 1  | 2    | 3           | LB   | GAP  | QUAL  | CPU |
| 1356.98.50.f  |  | 2,067.56 | 2839/2834 | 376/435.40   | 185/543,626.52   | 319/333.56   | 2/0.15   | 131 | 55 | 1100 | 199         | 2712 | 4.68 | 95.32 | 3.1 |
| - no R        |  | 2,069.80 | 2843/2836 | 376/435.40   | 183/4,000.52     | 321/335.00   | 0/0.00   | 131 | 55 | 1102 | 199         | 2712 | 4.83 | 95.17 | 3.0 |
| - shift       |  | 2,069.80 | -         | -            | 183/4,000.52     | 321/335.00   | -        | -   | -  | -    | -           | -    | -    | -     | 0.0 |
| 1356.126.40.f |  | 2,496.44 | 2836/2830 | 402/571.40   | 180/542,381.36   | 445/653.44   | 2/0.15   | 130 | 56 | 1106 | 190         | 2712 | 4.57 | 95.43 | 3.1 |
| - no R        |  | 2,499.72 | 2840/2832 | 402/571.40   | 178/2,755.36     | 447/655.20   | 0/0.00   | 130 | 56 | 1108 | 190         | 2712 | 4.72 | 95.28 | 3.2 |
| - shift       |  | 2,499.72 | -         | -            | 178/2,755.36     | 447/655.20   | -        | -   | -  | -    | -           | -    | -    | -     | 0.0 |
| 1356.161.30.f |  | 3,030.64 | 2855/2850 | 481/1,215.48 | 164/1,066,704.96 | 580/1,631.28 | 4/0.29   | 129 | 57 | 1120 | 197         | 2712 | 5.27 | 94.73 | 3.2 |
| - no R        |  | 3,040.48 | 2863/2857 | 482/1,231.40 | 160/2,850.96     | 585/1,659.24 | 0/0.00   | 129 | 57 | 1124 | 197         | 2712 | 5.57 | 94.43 | 3.2 |
| - shift       |  | 3,040.48 | -         | -            | 160/2,850.96     | 585/1,659.24 | -        | -   | -  | -    | -           | -    | -    | -     | 0.1 |
| 1356.128.50.f |  | 2,321.32 | 2845/2840 | 387/547.84   | 191/543,274.44   | 392/552.96   | 2/0.15   | 131 | 63 | 1100 | 197         | 2712 | 4.90 | 95.10 | 3.1 |
| - no R        |  | 2,324.12 | 2849/2842 | 387/547.84   | 189/3,642.92     | 394/554.40   | 0/0.00   | 131 | 63 | 1102 | 197         | 2712 | 5.05 | 94.95 | 3.1 |
| - shift       |  | 2,324.12 | -         | -            | 189/3,642.92     | 394/554.40   | -        | -   | -  | -    | -           | -    | -    | -     | 0.0 |
| 1356.160.40.f |  | 2,751.72 | 2823/2819 | 423/855.56   | 171/1,066,908.96 | 511/1,109.92 | 4/0.29   | 129 | 62 | 1103 | 177         | 2712 | 4.09 | 95.91 | 3.3 |
| - no R        |  | 2,760.60 | 2831/2826 | 424/865.36   | 167/3,051.96     | 516/1,125.80 | 0/0.00   | 129 | 62 | 1107 | 177         | 2712 | 4.39 | 95.61 | 3.1 |
| - shift       |  | 2,760.60 | -         | -            | 167/3,051.96     | 516/1,125.80 | -        | -   | -  | -    | -           | -    | -    | -     | 0.0 |
| 1356.216.30.f |  | 3,490.80 | 2819/2818 | 548/2,386.08 | 156/2,655,212.36 | 653/3,224.52 | 10/0.74  | 126 | 59 | 1116 | 172         | 2712 | 3.95 | 96.05 | 3.2 |
| - no R        |  | 3,516.72 | 2839/2834 | 551/2,523.76 | 144/3,167.24     | 666/3,349.72 | 0/0.00   | 126 | 60 | 1125 | 172         | 2712 | 4.68 | 95.32 | 3.3 |
| - shift       |  | 3,516.72 | -         | -            | 144/3,167.24     | 666/3,349.72 | -        | -   | -  | -    | -           | -    | -    | -     | 0.3 |

Table 5.49: 2 weeks instances solved with a strict forward search and  $\rho = 75\%$   $P$ .

| 2 weeks        |          | Solution  |              |                  |              |         | Criteria |    |      |     | Performance |      |       |     |
|----------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|------|-------|-----|
| $\rho = 100\%$ | T        | #M        | Wait         | Early            | Late         | R       | 0        | 1  | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| 1356.98.50     | 2,055.24 | 2830/2824 | 380/424.88   | 173/542,816.12   | 338/345.04   | 2/0.15  | 129      | 56 | 1111 | 180 | 2712        | 4.35 | 95.65 | 3.1 |
| - no R         | 2,057.16 | 2834/2826 | 380/424.88   | 171/3,190.12     | 340/346.32   | 0/0.00  | 129      | 56 | 1113 | 180 | 2712        | 4.50 | 95.50 | 3.1 |
| - shift        | 2,057.16 | -         | -            | 171/3,190.12     | 340/346.32   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.126.40    | 2,495.08 | 2837/2831 | 414/633.68   | 184/542,588.80   | 440/707.96   | 2/0.15  | 128      | 62 | 1112 | 181 | 2712        | 4.61 | 95.39 | 3.2 |
| - no R         | 2,498.28 | 2841/2833 | 414/633.68   | 182/2,960.04     | 442/709.64   | 0/0.00  | 128      | 62 | 1114 | 181 | 2712        | 4.76 | 95.24 | 3.1 |
| - shift        | 2,498.28 | -         | -            | 182/2,960.04     | 442/709.64   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.161.30    | 3,025.28 | 2832/2827 | 472/1,098.00 | 164/1,067,141.64 | 562/1,305.56 | 4/0.29  | 128      | 56 | 1117 | 179 | 2712        | 4.42 | 95.58 | 3.2 |
| - no R         | 3,034.32 | 2840/2834 | 473/1,110.40 | 160/3,287.64     | 567/1,330.24 | 0/0.00  | 128      | 56 | 1121 | 179 | 2712        | 4.72 | 95.28 | 3.2 |
| - shift        | 3,034.32 | -         | -            | 160/3,287.64     | 567/1,330.24 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.128.50    | 2,314.52 | 2836/2831 | 376/481.56   | 183/542,890.20   | 400/541.00   | 2/0.15  | 130      | 58 | 1115 | 179 | 2712        | 4.57 | 95.43 | 3.2 |
| - no R         | 2,317.48 | 2840/2833 | 376/481.56   | 181/3,261.28     | 402/542.60   | 0/0.00  | 130      | 58 | 1117 | 179 | 2712        | 4.72 | 95.28 | 3.2 |
| - shift        | 2,317.48 | -         | -            | 181/3,261.28     | 402/542.60   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.160.40    | 2,747.48 | 2837/2833 | 422/829.96   | 170/1,067,093.72 | 509/994.88   | 4/0.29  | 131      | 58 | 1123 | 173 | 2712        | 4.61 | 95.39 | 3.2 |
| - no R         | 2,756.28 | 2845/2840 | 424/839.52   | 166/3,236.80     | 514/1,010.36 | 0/0.00  | 131      | 58 | 1127 | 173 | 2712        | 4.90 | 95.10 | 3.2 |
| - shift        | 2,756.28 | -         | -            | 166/3,236.80     | 514/1,010.36 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.216.30    | 3,490.80 | 2819/2818 | 548/2,386.08 | 156/2,655,212.36 | 653/3,224.52 | 10/0.74 | 126      | 59 | 1116 | 172 | 2712        | 3.95 | 96.05 | 3.4 |
| - no R         | 3,516.72 | 2839/2834 | 551/2,523.76 | 144/3,167.24     | 666/3,349.72 | 0/0.00  | 126      | 60 | 1125 | 172 | 2712        | 4.68 | 95.32 | 3.3 |
| - shift        | 3,516.72 | -         | -            | 144/3,167.24     | 666/3,349.72 | -       | -        | -  | -    | -   | -           | -    | -     | 0.3 |

Table 5.50: 2 weeks instances solved with a combined forward and backward search and  $\rho = 100\%$   $P$ .

| 2 weeks (f)    |               | Solution |           |              |                  |              |         | Criteria |    |      |     | Performance |      |       |     |
|----------------|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|------|-------|-----|
|                |               | T        | #M        | Wait         | Early            | Late         | R       | 0        | 1  | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| $\rho = 100\%$ | 1356.98.50.f  | 2,058.36 | 2830/2825 | 371/391.00   | 168/542,644.96   | 347/373.72   | 2/0.15  | 128      | 58 | 1111 | 179 | 2712        | 4.35 | 95.65 | 3.1 |
|                | - no R        | 2,060.52 | 2834/2827 | 371/391.00   | 166/3,016.20     | 349/375.24   | 0/0.00  | 128      | 58 | 1113 | 179 | 2712        | 4.50 | 95.50 | 3.1 |
|                | - shift       | 2,060.52 | -         | -            | 166/3,016.20     | 349/375.24   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.126.40.f  | 1356.126.40.f | 2,495.16 | 2837/2831 | 412/624.00   | 180/542,574.76   | 442/706.80   | 2/0.15  | 128      | 62 | 1112 | 181 | 2712        | 4.61 | 95.39 | 3.1 |
|                | - no R        | 2,498.36 | 2841/2833 | 412/624.00   | 178/2,946.00     | 444/708.48   | 0/0.00  | 128      | 62 | 1114 | 181 | 2712        | 4.76 | 95.24 | 3.1 |
|                | - shift       | 2,498.36 | -         | -            | 178/2,946.00     | 444/708.48   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.161.30.f  | 1356.161.30.f | 3,025.28 | 2832/2827 | 472/1,098.00 | 164/1,067,141.64 | 562/1,305.56 | 4/0.29  | 128      | 56 | 1117 | 179 | 2712        | 4.42 | 95.58 | 3.3 |
|                | - no R        | 3,034.32 | 2840/2834 | 473/1,110.40 | 160/3,287.64     | 567/1,330.24 | 0/0.00  | 128      | 56 | 1121 | 179 | 2712        | 4.72 | 95.28 | 3.2 |
|                | - shift       | 3,034.32 | -         | -            | 160/3,287.64     | 567/1,330.24 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.128.50.f  | 1356.128.50.f | 2,314.04 | 2836/2831 | 375/471.32   | 181/542,878.80   | 399/520.36   | 2/0.15  | 130      | 58 | 1115 | 179 | 2712        | 4.57 | 95.43 | 3.2 |
|                | - no R        | 2,317.00 | 2840/2833 | 375/471.32   | 179/3,249.88     | 401/521.96   | 0/0.00  | 130      | 58 | 1117 | 179 | 2712        | 4.72 | 95.28 | 3.1 |
|                | - shift       | 2,317.00 | -         | -            | 179/3,249.88     | 401/521.96   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.160.40.f  | 1356.160.40.f | 2,747.48 | 2837/2833 | 422/829.96   | 170/1,067,093.72 | 509/994.88   | 4/0.29  | 131      | 58 | 1123 | 173 | 2712        | 4.61 | 95.39 | 3.3 |
|                | - no R        | 2,756.28 | 2845/2840 | 424/839.52   | 166/3,236.80     | 514/1,010.36 | 0/0.00  | 131      | 58 | 1127 | 173 | 2712        | 4.90 | 95.10 | 3.2 |
|                | - shift       | 2,756.28 | -         | -            | 166/3,236.80     | 514/1,010.36 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1356.216.30.f  | 1356.216.30.f | 3,490.80 | 2819/2818 | 548/2,386.08 | 156/2,655,212.36 | 653/3,224.52 | 10/0.74 | 126      | 59 | 1116 | 172 | 2712        | 3.95 | 96.05 | 3.4 |
|                | - no R        | 3,516.72 | 2839/2834 | 551/2,523.76 | 144/3,167.24     | 666/3,349.72 | 0/0.00  | 126      | 60 | 1125 | 172 | 2712        | 4.68 | 95.32 | 3.3 |
|                | - shift       | 3,516.72 | -         | -            | 144/3,167.24     | 666/3,349.72 | -       | -        | -  | -    | -   | -           | -    | -     | 0.3 |

Table 5.51: 2 weeks instances solved with a strict forward search and  $\rho = 100\%$   $P$ .



| 3 weeks       |          |           |              |                  |              |          |     |    |      | Criteria |             |       |       |     | Performance |  |  |  |
|---------------|----------|-----------|--------------|------------------|--------------|----------|-----|----|------|----------|-------------|-------|-------|-----|-------------|--|--|--|
| $\rho = 25\%$ | Solution |           |              |                  |              | Criteria |     |    |      |          | Performance |       |       |     |             |  |  |  |
|               | T        | #M        | Wait         | Early            | Late         | R        | 0   | 1  | 2    | 3        | LB          | GAP   | QUAL  | CPU |             |  |  |  |
| 1860.98.50    | 2,980.36 | 4505/4496 | 594/915.76   | 270/1,619,969.56 | 515/875.12   | 6/0.32   | 185 | 64 | 1670 | 732      | 3720        | 21.10 | 78.90 | 5.4 |             |  |  |  |
| - no R        | 2,988.36 | 4562/4552 | 608/1,007.88 | 256/7,687.52     | 543/964.40   | 0/0.00   | 188 | 60 | 1683 | 771      | 3720        | 22.63 | 77.37 | 5.3 |             |  |  |  |
| - shift       | 2,988.36 | -         | -            | 256/7,687.52     | 543/964.40   | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 |             |  |  |  |
| 1860.126.40   | 3,531.48 | 4329/4325 | 680/1,463.40 | 264/1,620,194.04 | 645/1,483.28 | 6/0.32   | 170 | 75 | 1625 | 605      | 3720        | 16.37 | 83.63 | 5.5 |             |  |  |  |
| - no R        | 3,541.48 | 4345/4335 | 680/1,476.84 | 258/7,941.96     | 652/1,520.36 | 0/0.00   | 170 | 77 | 1630 | 608      | 3720        | 16.80 | 83.20 | 5.3 |             |  |  |  |
| - shift       | 3,541.48 | -         | -            | 258/7,941.96     | 652/1,520.36 | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 |             |  |  |  |
| 1860.161.30   | 4,230.00 | 4213/4211 | 747/2,397.04 | 243/2,426,208.12 | 798/2,784.56 | 9/0.48   | 167 | 75 | 1589 | 531      | 3720        | 13.25 | 86.75 | 5.4 |             |  |  |  |
| - no R        | 4,268.08 | 4257/4248 | 743/2,351.88 | 237/7,426.40     | 811/2,654.20 | 0/0.00   | 170 | 70 | 1607 | 550      | 3720        | 14.44 | 85.56 | 5.4 |             |  |  |  |
| - shift       | 4,268.08 | -         | -            | 237/7,426.40     | 811/2,654.20 | -        | -   | -  | -    | -        | -           | -     | -     | 0.1 |             |  |  |  |
| 1860.128.50   | 3,345.40 | 4416/4412 | 616/1,184.76 | 270/1,617,694.96 | 584/1,172.44 | 6/0.32   | 181 | 70 | 1641 | 670      | 3720        | 18.71 | 81.29 | 5.3 |             |  |  |  |
| - no R        | 3,354.12 | 4430/4420 | 616/1,194.04 | 264/7,063.28     | 593/1,188.76 | 0/0.00   | 181 | 70 | 1647 | 672      | 3720        | 19.09 | 80.91 | 5.3 |             |  |  |  |
| - shift       | 3,354.12 | -         | -            | 264/7,063.28     | 593/1,188.76 | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 |             |  |  |  |
| 1860.160.40   | 3,897.56 | 4304/4301 | 684/2,045.16 | 265/1,619,211.44 | 741/2,049.96 | 6/0.32   | 177 | 68 | 1606 | 599      | 3720        | 15.70 | 84.30 | 5.4 |             |  |  |  |
| - no R        | 3,899.48 | 4339/4330 | 705/1,944.36 | 263/7,487.64     | 726/1,897.76 | 0/0.00   | 175 | 75 | 1612 | 617      | 3720        | 16.64 | 83.36 | 5.3 |             |  |  |  |
| - shift       | 3,899.48 | -         | -            | 263/7,487.64     | 726/1,897.76 | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 |             |  |  |  |
| 1860.216.30   | 4,867.20 | 4062/4060 | 817/3,656.92 | 225/3,773,521.60 | 926/4,148.36 | 14/0.75  | 168 | 65 | 1566 | 417      | 3720        | 9.19  | 90.81 | 5.4 |             |  |  |  |
| - no R        | 4,901.76 | 4079/4071 | 830/3,731.72 | 207/4,474.08     | 928/4,231.72 | 0/0.00   | 164 | 70 | 1588 | 397      | 3720        | 9.65  | 90.35 | 5.4 |             |  |  |  |
| - shift       | 4,901.76 | -         | -            | 207/4,474.08     | 928/4,231.72 | -        | -   | -  | -    | -        | -           | -     | -     | 0.3 |             |  |  |  |

Table 5.52: 3 weeks instances solved with a combined forward and backward search and  $\rho = 25\%$   $P$ .

| 3 weeks (f)   |          | Solution  |              |                  |              |         | Criteria |    |      |     | Performance |       |       |      |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|-------|-------|------|
|               |          | T         | #M           | Wait             | Early        | Late    | R        | 0  | 1    | 2   | 3           | LB    | GAP   | QUAL |
| 1860.98.50.f  | 2,989.32 | 4542/4533 | 597/911.88   | 271/1,112,151.36 | 517/859.72   | 4/0.22  | 169      | 59 | 1687 | 771 | 3720        | 22.10 | 77.90 | 5.2  |
| - no R        | 2,972.76 | 4503/4490 | 595/918.68   | 276/8,812.08     | 517/824.04   | 0/0.00  | 186      | 63 | 1654 | 740 | 3720        | 21.05 | 78.95 | 5.3  |
| - shift       | 2,972.76 | -         | -            | 276/8,812.08     | 517/824.04   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.126.40.f | 3,531.96 | 4329/4325 | 676/1,453.40 | 264/1,620,189.76 | 646/1,524.68 | 6/0.32  | 170      | 76 | 1624 | 605 | 3720        | 16.37 | 83.63 | 5.4  |
| - no R        | 3,541.96 | 4345/4335 | 676/1,466.84 | 258/7,937.68     | 653/1,561.76 | 0/0.00  | 170      | 78 | 1629 | 608 | 3720        | 16.80 | 83.20 | 5.3  |
| - shift       | 3,541.96 | -         | -            | 258/7,937.68     | 653/1,561.76 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.161.30.f | 4,230.00 | 4213/4211 | 747/2,397.04 | 243/2,426,208.12 | 798/2,785.84 | 9/0.48  | 167      | 75 | 1589 | 531 | 3720        | 13.25 | 86.75 | 5.3  |
| - no R        | 4,267.84 | 4257/4248 | 742/2,347.48 | 237/7,426.40     | 812/2,663.00 | 0/0.00  | 170      | 70 | 1607 | 550 | 3720        | 14.44 | 85.56 | 5.3  |
| - shift       | 4,267.84 | -         | -            | 237/7,426.40     | 812/2,663.00 | -       | -        | -  | -    | -   | -           | -     | -     | 0.1  |
| 1860.128.50.f | 3,344.92 | 4416/4412 | 614/1,179.52 | 270/1,617,700.04 | 583/1,164.88 | 6/0.32  | 181      | 70 | 1641 | 670 | 3720        | 18.71 | 81.29 | 5.3  |
| - no R        | 3,353.64 | 4430/4420 | 614/1,188.80 | 264/7,068.36     | 592/1,181.20 | 0/0.00  | 181      | 70 | 1647 | 672 | 3720        | 19.09 | 80.91 | 5.3  |
| - shift       | 3,353.64 | -         | -            | 264/7,068.36     | 592/1,181.20 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.160.40.f | 3,897.56 | 4304/4301 | 684/2,045.16 | 265/1,619,211.44 | 741/2,049.96 | 6/0.32  | 177      | 68 | 1606 | 599 | 3720        | 15.70 | 84.30 | 5.4  |
| - no R        | 3,898.60 | 4339/4330 | 705/1,944.36 | 263/7,401.00     | 726/1,895.84 | 0/0.00  | 175      | 75 | 1612 | 617 | 3720        | 16.64 | 83.36 | 5.4  |
| - shift       | 3,898.60 | -         | -            | 263/7,401.00     | 726/1,895.84 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.216.30.f | 4,867.20 | 4062/4060 | 817/3,656.92 | 225/3,773,521.60 | 926/4,148.36 | 14/0.75 | 168      | 65 | 1566 | 417 | 3720        | 9.19  | 90.81 | 5.3  |
| - no R        | 4,901.76 | 4079/4071 | 830/3,731.72 | 207/4,474.08     | 928/4,231.72 | 0/0.00  | 164      | 70 | 1588 | 397 | 3720        | 9.65  | 90.35 | 5.4  |
| - shift       | 4,901.76 | -         | -            | 207/4,474.08     | 928/4,231.72 | -       | -        | -  | -    | -   | -           | -     | -     | 0.3  |

Table 5.53: 3 weeks instances solved with a strict forward search and  $\rho = 25\%$   $P$ .

| 3 weeks       |          | Solution  |              |                  |              |         |     |    |      |     |      | Criteria |       |      | Performance |  |  |  |
|---------------|----------|-----------|--------------|------------------|--------------|---------|-----|----|------|-----|------|----------|-------|------|-------------|--|--|--|
| $\rho = 33\%$ |          | T         | #M           | Wait             | Early        | Late    | R   | 0  | 1    | 2   | 3    | LB       | GAP   | QUAL | CPU         |  |  |  |
| 1860.98.50    | 2,934.36 | 4382/4378 | 580/787.76   | 255/1,372,843.20 | 510/831.84   | 5/0.27  | 166 | 66 | 1651 | 644 | 3720 | 17.80    | 82.20 | 5.3  |             |  |  |  |
| - no R        | 2,939.72 | 4399/4390 | 583/803.24   | 250/6,854.44     | 514/841.76   | 0/0.00  | 166 | 70 | 1655 | 648 | 3720 | 18.25    | 81.75 | 5.3  |             |  |  |  |
| - shift       | 2,939.72 | -         | -            | 250/6,854.44     | 514/841.76   | -       | -   | -  | -    | -   | -    | -        | -     | 0.0  |             |  |  |  |
| 1860.126.40   | 3,503.16 | 4216/4212 | 616/1,182.44 | 259/1,634,766.44 | 616/1,167.48 | 6/0.32  | 169 | 65 | 1604 | 524 | 3720 | 13.33    | 86.67 | 5.5  |             |  |  |  |
| - no R        | 3,512.28 | 4228/4220 | 620/1,193.32 | 253/6,753.52     | 624/1,186.44 | 0/0.00  | 169 | 66 | 1609 | 524 | 3720 | 13.66    | 86.34 | 5.3  |             |  |  |  |
| - shift       | 3,512.28 | -         | -            | 253/6,753.52     | 624/1,186.44 | -       | -   | -  | -    | -   | -    | -        | -     | 0.0  |             |  |  |  |
| 1860.161.30   | 4,188.24 | 4006/4004 | 719/1,920.96 | 228/2,163,058.44 | 801/2,283.48 | 8/0.43  | 172 | 71 | 1538 | 373 | 3720 | 7.69     | 92.31 | 5.4  |             |  |  |  |
| - no R        | 4,204.56 | 4022/4015 | 720/1,942.00 | 220/5,365.60     | 809/2,320.56 | 0/0.00  | 171 | 72 | 1546 | 373 | 3720 | 8.12     | 91.88 | 5.4  |             |  |  |  |
| - shift       | 4,204.56 | -         | -            | 220/5,365.60     | 809/2,320.56 | -       | -   | -  | -    | -   | -    | -        | -     | 0.1  |             |  |  |  |
| 1860.128.50   | 3,303.00 | 4285/4281 | 612/1,090.44 | 260/833,016.80   | 599/1,035.44 | 3/0.16  | 173 | 64 | 1615 | 576 | 3720 | 15.19    | 84.81 | 5.4  |             |  |  |  |
| - no R        | 3,307.32 | 4291/4284 | 612/1,090.44 | 257/7,113.80     | 602/1,037.96 | 0/0.00  | 173 | 64 | 1618 | 576 | 3720 | 15.35    | 84.65 | 5.3  |             |  |  |  |
| - shift       | 3,307.32 | -         | -            | 257/7,113.80     | 602/1,037.96 | -       | -   | -  | -    | -   | -    | -        | -     | 0.0  |             |  |  |  |
| 1860.160.40   | 3,829.32 | 4081/4076 | 668/1,783.24 | 238/1,615,758.32 | 727/1,895.72 | 6/0.32  | 171 | 83 | 1563 | 410 | 3720 | 9.70     | 90.30 | 5.4  |             |  |  |  |
| - no R        | 3,840.68 | 4093/4085 | 669/1,795.00 | 232/5,129.60     | 735/1,919.96 | 0/0.00  | 171 | 84 | 1568 | 410 | 3720 | 10.03    | 89.97 | 5.3  |             |  |  |  |
| - shift       | 3,840.68 | -         | -            | 232/5,129.60     | 735/1,919.96 | -       | -   | -  | -    | -   | -    | -        | -     | 0.1  |             |  |  |  |
| 1860.216.30   | 4,831.92 | 3909/3905 | 820/3,252.52 | 242/2,699,565.00 | 880/3,814.20 | 10/0.54 | 171 | 90 | 1515 | 283 | 3720 | 5.08     | 94.92 | 5.6  |             |  |  |  |
| - no R        | 4,856.40 | 3929/3920 | 823/3,337.76 | 232/5,063.92     | 894/3,900.84 | 0/0.00  | 171 | 90 | 1525 | 283 | 3720 | 5.62     | 94.38 | 5.4  |             |  |  |  |
| - shift       | 4,856.40 | -         | -            | 232/5,063.92     | 894/3,900.84 | -       | -   | -  | -    | -   | -    | -        | -     | 0.3  |             |  |  |  |

Table 5.54: 3 weeks instances solved with a combined forward and backward search and  $\rho = 33\%$   $P$ .

| 3 weeks (f)   |          |           |              |                  |              |         |          |    |      |     |             |       |       |     |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|-------|-------|-----|
| $\rho = 33\%$ | Solution |           |              |                  |              |         | Criteria |    |      |     | Performance |       |       |     |
|               | T        | #M        | Wait         | Early            | Late         | R       | 0        | 1  | 2    | 3   | LB          | GAP   | QUAL  | CPU |
| 1860.98.50.f  | 2,938.52 | 4334/4332 | 571/777.52   | 262/1,642,787.32 | 495/706.28   | 6/0.32  | 179      | 75 | 1624 | 602 | 3720        | 16.51 | 83.49 | 5.3 |
| - no R        | 2,944.68 | 4339/4331 | 576/767.36   | 239/6,717.24     | 522/785.56   | 0/0.00  | 179      | 66 | 1642 | 592 | 3720        | 16.64 | 83.36 | 5.4 |
| - shift       | 2,944.68 | -         | -            | 239/6,717.24     | 522/785.56   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1860.126.40.f | 3,503.08 | 4216/4212 | 616/1,181.24 | 259/1,634,753.32 | 615/1,154.52 | 6/0.32  | 169      | 65 | 1604 | 524 | 3720        | 13.33 | 86.67 | 5.4 |
| - no R        | 3,512.20 | 4228/4220 | 620/1,192.12 | 253/6,740.40     | 623/1,173.48 | 0/0.00  | 169      | 66 | 1609 | 524 | 3720        | 13.66 | 86.34 | 5.6 |
| - shift       | 3,512.20 | -         | -            | 253/6,740.40     | 623/1,173.48 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1860.161.30.f | 4,188.24 | 4006/4004 | 719/1,920.96 | 228/2,163,058.44 | 801/2,283.48 | 8/0.43  | 172      | 71 | 1538 | 373 | 3720        | 7.69  | 92.31 | 5.4 |
| - no R        | 4,204.56 | 4022/4015 | 720/1,942.00 | 220/5,365.60     | 809/2,320.56 | 0/0.00  | 171      | 72 | 1546 | 373 | 3720        | 8.12  | 91.88 | 5.5 |
| - shift       | 4,204.56 | -         | -            | 220/5,365.60     | 809/2,320.56 | -       | -        | -  | -    | -   | -           | -     | -     | 0.1 |
| 1860.128.50.f | 3,303.40 | 4285/4281 | 611/1,082.88 | 260/833,019.12   | 600/1,033.20 | 3/0.16  | 173      | 64 | 1615 | 576 | 3720        | 15.19 | 84.81 | 5.3 |
| - no R        | 3,307.72 | 4291/4284 | 611/1,082.88 | 257/7,116.12     | 603/1,035.72 | 0/0.00  | 173      | 64 | 1618 | 576 | 3720        | 15.35 | 84.65 | 5.8 |
| - shift       | 3,307.72 | -         | -            | 257/7,116.12     | 603/1,035.72 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 1860.160.40.f | 3,829.32 | 4081/4076 | 668/1,783.24 | 238/1,615,758.32 | 727/1,895.72 | 6/0.32  | 171      | 83 | 1563 | 410 | 3720        | 9.70  | 90.30 | 5.4 |
| - no R        | 3,840.68 | 4093/4085 | 669/1,795.00 | 232/5,129.60     | 735/1,919.96 | 0/0.00  | 171      | 84 | 1568 | 410 | 3720        | 10.03 | 89.97 | 5.7 |
| - shift       | 3,840.68 | -         | -            | 232/5,129.60     | 735/1,919.96 | -       | -        | -  | -    | -   | -           | -     | -     | 0.1 |
| 1860.216.30.f | 4,831.92 | 3909/3905 | 820/3,252.52 | 242/2,699,565.00 | 880/3,814.20 | 10/0.54 | 171      | 90 | 1515 | 283 | 3720        | 5.08  | 94.92 | 5.4 |
| - no R        | 4,856.40 | 3929/3920 | 823/3,337.76 | 232/5,063.92     | 894/3,900.84 | 0/0.00  | 171      | 90 | 1525 | 283 | 3720        | 5.62  | 94.38 | 5.8 |
| - shift       | 4,856.40 | -         | -            | 232/5,063.92     | 894/3,900.84 | -       | -        | -  | -    | -   | -           | -     | -     | 0.3 |

Table 5.55: 3 weeks instances solved with a strict forward search and  $\rho = 33\%$   $P$ .

| 3 weeks       |          |           |              |                  |              |         |     |    |      |          |      |       |             |     |  |
|---------------|----------|-----------|--------------|------------------|--------------|---------|-----|----|------|----------|------|-------|-------------|-----|--|
| $\rho = 50\%$ |          |           |              |                  |              |         |     |    |      |          |      |       |             |     |  |
| Solution      |          |           |              |                  |              |         |     |    |      | Criteria |      |       | Performance |     |  |
| T             | #M       | Wait      | Early        | Late             | R            | 0       | 1   | 2  | 3    | LB       | GAP  | QUAL  | CPU         |     |  |
| 1860.98.50    | 2,892.04 | 4153/4146 | 554/700.16   | 255/1,094,184.72 | 491/634.44   | 4/0.22  | 161 | 58 | 1601 | 477      | 3720 | 11.64 | 88.36       | 5.4 |  |
| - no R        | 2,896.84 | 4161/4150 | 555/701.64   | 252/6,164.80     | 495/638.20   | 0/0.00  | 161 | 59 | 1604 | 477      | 3720 | 11.85 | 88.15       | 5.4 |  |
| - shift       | 2,896.84 | -         | -            | 252/6,164.80     | 495/638.20   | -       | -   | -  | -    | -        | -    | -     | -           | 0.0 |  |
| 1860.126.40   | 3,427.88 | 3889/3882 | 594/842.60   | 249/1,092,753.28 | 604/837.76   | 4/0.22  | 169 | 75 | 1522 | 267      | 3720 | 4.54  | 95.46       | 5.7 |  |
| - no R        | 3,433.96 | 3897/3887 | 595/845.88   | 245/4,733.36     | 609/844.12   | 0/0.00  | 169 | 76 | 1525 | 267      | 3720 | 4.76  | 95.24       | 5.4 |  |
| - shift       | 3,433.96 | -         | -            | 245/4,733.36     | 609/844.12   | -       | -   | -  | -    | -        | -    | -     | -           | 0.0 |  |
| 1860.161.30   | 4,154.32 | 3871/3867 | 688/1,640.64 | 226/1,632,441.92 | 779/1,905.20 | 6/0.32  | 167 | 82 | 1535 | 233      | 3720 | 4.06  | 95.94       | 5.5 |  |
| - no R        | 4,167.60 | 3883/3876 | 688/1,665.36 | 220/4,428.88     | 786/1,946.68 | 0/0.00  | 167 | 82 | 1541 | 233      | 3720 | 4.38  | 95.62       | 5.5 |  |
| - shift       | 4,167.60 | -         | -            | 220/4,428.88     | 786/1,946.68 | -       | -   | -  | -    | -        | -    | -     | -           | 0.1 |  |
| 1860.128.50   | 3,207.00 | 3931/3925 | 551/797.60   | 271/832,001.68   | 575/884.32   | 3/0.16  | 172 | 81 | 1521 | 300      | 3720 | 5.67  | 94.33       | 5.4 |  |
| - no R        | 3,211.40 | 3937/3928 | 551/797.60   | 268/6,095.76     | 578/886.76   | 0/0.00  | 172 | 81 | 1524 | 300      | 3720 | 5.83  | 94.17       | 5.3 |  |
| - shift       | 3,211.40 | -         | -            | 268/6,095.76     | 578/886.76   | -       | -   | -  | -    | -        | -    | -     | -           | 0.0 |  |
| 1860.160.40   | 3,767.40 | 3896/3892 | 623/1,206.56 | 242/1,354,172.52 | 699/1,338.44 | 5/0.27  | 167 | 79 | 1538 | 257      | 3720 | 4.73  | 95.27       | 5.4 |  |
| - no R        | 3,776.52 | 3906/3899 | 624/1,216.28 | 237/4,038.60     | 704/1,352.92 | 0/0.00  | 167 | 79 | 1543 | 257      | 3720 | 5.00  | 95.00       | 5.5 |  |
| - shift       | 3,776.52 | -         | -            | 237/4,038.60     | 704/1,352.92 | -       | -   | -  | -    | -        | -    | -     | -           | 0.0 |  |
| 1860.216.30   | 4,802.40 | 3857/3856 | 810/3,212.76 | 231/3,230,479.16 | 882/4,035.92 | 12/0.65 | 166 | 65 | 1535 | 243      | 3720 | 3.68  | 96.32       | 5.5 |  |
| - no R        | 4,832.16 | 3881/3874 | 814/3,352.44 | 219/4,691.76     | 895/4,142.76 | 0/0.00  | 166 | 66 | 1546 | 243      | 3720 | 4.33  | 95.67       | 5.7 |  |
| - shift       | 4,832.16 | -         | -            | 219/4,691.76     | 895/4,142.76 | -       | -   | -  | -    | -        | -    | -     | -           | 0.3 |  |

Table 5.56: 3 weeks instances solved with a combined forward and backward search and  $\rho = 50\%$   $P$ .

| 3 weeks (f)   |          | Solution  |              |                  |              |         | Criteria |    |      |     | Performance |       |       |      |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|-------|-------|------|
|               |          | T         | #M           | Wait             | Early        | Late    | R        | 0  | 1    | 2   | 3           | LB    | GAP   | QUAL |
| 1860.98.50.f  | 2,890.60 | 4151/4142 | 554/693.72   | 268/1,094,512.56 | 469/570.96   | 4/0.22  | 164      | 71 | 1589 | 471 | 3720        | 11.59 | 88.41 | 5.4  |
| - no R        | 2,895.40 | 4159/4147 | 555/695.40   | 264/6,491.56     | 473/575.32   | 0/0.00  | 164      | 71 | 1593 | 471 | 3720        | 11.80 | 88.20 | 5.4  |
| - shift       | 2,895.40 | -         | -            | 264/6,491.56     | 473/575.32   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.126.40.f | 3,428.12 | 3889/3882 | 594/839.24   | 248/1,092,752.32 | 605/843.00   | 4/0.22  | 169      | 75 | 1522 | 267 | 3720        | 4.54  | 95.46 | 5.4  |
| - no R        | 3,434.20 | 3897/3887 | 595/842.52   | 244/4,732.40     | 610/849.36   | 0/0.00  | 169      | 76 | 1525 | 267 | 3720        | 4.76  | 95.24 | 5.5  |
| - shift       | 3,434.20 | -         | -            | 244/4,732.40     | 610/849.36   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.161.30.f | 4,154.32 | 3871/3867 | 688/1,640.64 | 226/1,632,441.92 | 779/1,905.20 | 6/0.32  | 167      | 82 | 1535 | 233 | 3720        | 4.06  | 95.94 | 5.4  |
| - no R        | 4,167.60 | 3883/3876 | 688/1,665.36 | 220/4,428.88     | 786/1,946.68 | 0/0.00  | 167      | 82 | 1541 | 233 | 3720        | 4.38  | 95.62 | 5.6  |
| - shift       | 4,167.60 | -         | -            | 220/4,428.88     | 786/1,946.68 | -       | -        | -  | -    | -   | -           | -     | -     | 0.1  |
| 1860.128.50.f | 3,206.84 | 3931/3925 | 551/797.04   | 270/831,986.40   | 576/878.24   | 3/0.16  | 172      | 81 | 1521 | 300 | 3720        | 5.67  | 94.33 | 5.4  |
| - no R        | 3,211.24 | 3937/3928 | 551/797.04   | 267/6,080.48     | 579/880.68   | 0/0.00  | 172      | 81 | 1524 | 300 | 3720        | 5.83  | 94.17 | 5.4  |
| - shift       | 3,211.24 | -         | -            | 267/6,080.48     | 579/880.68   | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.160.40.f | 3,767.40 | 3896/3892 | 623/1,206.56 | 242/1,354,172.52 | 699/1,338.44 | 5/0.27  | 167      | 79 | 1538 | 257 | 3720        | 4.73  | 95.27 | 5.4  |
| - no R        | 3,776.52 | 3906/3899 | 624/1,216.28 | 237/4,038.60     | 704/1,352.92 | 0/0.00  | 167      | 79 | 1543 | 257 | 3720        | 5.00  | 95.00 | 5.5  |
| - shift       | 3,776.52 | -         | -            | 237/4,038.60     | 704/1,352.92 | -       | -        | -  | -    | -   | -           | -     | -     | 0.0  |
| 1860.216.30.f | 4,802.40 | 3857/3856 | 810/3,212.76 | 231/3,230,479.16 | 882/4,035.92 | 12/0.65 | 166      | 65 | 1535 | 243 | 3720        | 3.68  | 96.32 | 5.5  |
| - no R        | 4,832.16 | 3881/3874 | 814/3,352.44 | 219/4,691.76     | 895/4,142.76 | 0/0.00  | 166      | 66 | 1546 | 243 | 3720        | 4.33  | 95.67 | 5.5  |
| - shift       | 4,832.16 | -         | -            | 219/4,691.76     | 895/4,142.76 | -       | -        | -  | -    | -   | -           | -     | -     | 0.3  |

Table 5.57: 3 weeks instances solved with a strict forward search and  $\rho = 50\%$   $P$ .

| 3 weeks       |          |           | Solution     |                  |              |         | Criteria |    |      |     | Performance |      |       |     |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|------|-------|-----|
| $\rho = 67\%$ | T        | #M        | Wait         | Early            | Late         | R       | 0        | 1  | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| 1860.98.50    | 2,829.88 | 3927/3922 | 544/617.32   | 233/830.669.76   | 490/562.44   | 3/0.16  | 173      | 67 | 1553 | 277 | 3720        | 5.56 | 94.44 | 5.5 |
| - no R        | 2,832.92 | 3933/3925 | 544/617.32   | 230/4,756.44     | 493/564.72   | 0/0.00  | 173      | 67 | 1556 | 277 | 3720        | 5.73 | 94.27 | 5.4 |
| - shift       | 2,832.92 | -         | -            | 230/4,756.44     | 493/564.72   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.126.40   | 3,426.28 | 3892/3886 | 595/894.84   | 237/829.697.56   | 610/901.20   | 3/0.16  | 167      | 78 | 1531 | 259 | 3720        | 4.62 | 95.38 | 5.5 |
| - no R        | 3,430.68 | 3898/3889 | 595/894.84   | 234/3,783.68     | 613/903.88   | 0/0.00  | 167      | 78 | 1534 | 259 | 3720        | 4.78 | 95.22 | 5.6 |
| - shift       | 3,430.68 | -         | -            | 234/3,783.68     | 613/903.88   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.161.30   | 4,157.44 | 3909/3904 | 698/1,836.96 | 232/1,632,159.48 | 777/2,011.88 | 6/0.32  | 166      | 83 | 1541 | 265 | 3720        | 5.08 | 94.92 | 5.5 |
| - no R        | 4,171.12 | 3921/3913 | 699/1,859.76 | 226/4,149.48     | 785/2,049.72 | 0/0.00  | 166      | 83 | 1547 | 265 | 3720        | 5.40 | 94.60 | 5.5 |
| - shift       | 4,171.12 | -         | -            | 226/4,149.48     | 785/2,049.72 | -       | -        | -  | -    | -   | -           | -    | -     | 0.1 |
| 1860.128.50   | 3,173.56 | 3887/3881 | 545/742.24   | 243/830,020.88   | 569/760.92   | 3/0.16  | 168      | 72 | 1536 | 254 | 3720        | 4.49 | 95.51 | 5.4 |
| - no R        | 3,177.88 | 3893/3884 | 545/742.24   | 240/4,117.88     | 572/763.36   | 0/0.00  | 168      | 72 | 1539 | 254 | 3720        | 4.65 | 95.35 | 5.5 |
| - shift       | 3,177.88 | -         | -            | 240/4,117.88     | 572/763.36   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.160.40   | 3,769.72 | 3879/3875 | 624/1,312.36 | 234/1,632,524.00 | 698/1,416.84 | 6/0.32  | 166      | 82 | 1535 | 242 | 3720        | 4.27 | 95.73 | 5.5 |
| - no R        | 3,782.28 | 3891/3884 | 628/1,329.96 | 228/4,511.00     | 706/1,443.00 | 0/0.00  | 166      | 82 | 1541 | 242 | 3720        | 4.60 | 95.40 | 5.4 |
| - shift       | 3,782.28 | -         | -            | 228/4,511.00     | 706/1,443.00 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.216.30   | 4,784.88 | 3852/3851 | 805/3,126.24 | 213/3,502,389.40 | 886/3,976.28 | 13/0.70 | 165      | 83 | 1535 | 222 | 3720        | 3.55 | 96.45 | 5.6 |
| - no R        | 4,818.00 | 3878/3871 | 810/3,278.36 | 198/4,407.28     | 903/4,119.60 | 0/0.00  | 165      | 84 | 1547 | 222 | 3720        | 4.25 | 95.75 | 5.6 |
| - shift       | 4,818.00 | -         | -            | 198/4,407.28     | 903/4,119.60 | -       | -        | -  | -    | -   | -           | -    | -     | 0.3 |

Table 5.58: 3 weeks instances solved with a combined forward and backward search and  $\rho = 67\%$   $P$ .

| 3 weeks (f) |  |  |  |  |  |  |  |  |  | Criteria |  |  |  | Performance |  |  |  |
|-------------|--|--|--|--|--|--|--|--|--|----------|--|--|--|-------------|--|--|--|
| Solution    |  |  |  |  |  |  |  |  |  | Criteria |  |  |  | Performance |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |
|             |  |  |  |  |  |  |  |  |  |          |  |  |  |             |  |  |  |



| 3 weeks       |          |           |              |                  |              |         |     |    |      |          |      |      |             |     |
|---------------|----------|-----------|--------------|------------------|--------------|---------|-----|----|------|----------|------|------|-------------|-----|
| $\rho = 75\%$ |          |           |              |                  |              |         |     |    |      |          |      |      |             |     |
| Solution      |          |           |              |                  |              |         |     |    |      | Criteria |      |      | Performance |     |
|               | T        | #M        | Wait         | Early            | Late         | R       | 0   | 1  | 2    | 3        | LB   | GAP  | QUAL        | CPU |
| 1860.98.50    | 2,835.24 | 3882/3876 | 547/651.88   | 232/1,092,120.96 | 456/468.72   | 4/0.22  | 169 | 77 | 1535 | 245      | 3720 | 4.35 | 95.65       | 5.6 |
| - no R        | 2,840.12 | 3890/3882 | 548/655.36   | 228/4,103.96     | 461/474.60   | 0/0.00  | 169 | 77 | 1539 | 245      | 3720 | 4.57 | 95.43       | 5.5 |
| - shift       | 2,840.12 | -         | -            | 228/4,103.96     | 461/474.60   | -       | -   | -  | -    | -        | -    | -    | -           | 0.0 |
| 1860.126.40   | 3,427.48 | 3901/3895 | 602/940.68   | 229/1,091,615.56 | 621/913.08   | 4/0.22  | 168 | 77 | 1553 | 247      | 3720 | 4.87 | 95.13       | 5.4 |
| - no R        | 3,433.72 | 3909/3900 | 603/943.24   | 225/3,598.56     | 626/918.72   | 0/0.00  | 168 | 78 | 1556 | 247      | 3720 | 5.08 | 94.92       | 5.6 |
| - shift       | 3,433.72 | -         | -            | 225/3,598.56     | 626/918.72   | -       | -   | -  | -    | -        | -    | -    | -           | 0.0 |
| 1860.161.30   | 4,157.44 | 3909/3904 | 698/1,836.96 | 232/1,632,159.48 | 777/2,011.88 | 6/0.32  | 166 | 83 | 1541 | 265      | 3720 | 5.08 | 94.92       | 5.5 |
| - no R        | 4,171.12 | 3921/3913 | 699/1,859.76 | 226/4,149.48     | 785/2,049.72 | 0/0.00  | 166 | 83 | 1547 | 265      | 3720 | 5.40 | 94.60       | 5.8 |
| - shift       | 4,171.12 | -         | -            | 226/4,149.48     | 785/2,049.72 | -       | -   | -  | -    | -        | -    | -    | -           | 0.1 |
| 1860.128.50   | 3,194.20 | 3902/3896 | 569/804.48   | 253/830,674.72   | 546/771.44   | 3/0.16  | 170 | 86 | 1531 | 258      | 3720 | 4.89 | 95.11       | 5.4 |
| - no R        | 3,198.36 | 3908/3899 | 569/804.48   | 250/4,766.20     | 549/773.80   | 0/0.00  | 170 | 86 | 1534 | 258      | 3720 | 5.05 | 94.95       | 5.6 |
| - shift       | 3,198.36 | -         | -            | 250/4,766.20     | 549/773.80   | -       | -   | -  | -    | -        | -    | -    | -           | 0.0 |
| 1860.160.40   | 3,770.60 | 3879/3875 | 624/1,312.36 | 234/1,632,524.00 | 698/1,418.76 | 6/0.32  | 166 | 82 | 1535 | 242      | 3720 | 4.27 | 95.73       | 5.5 |
| - no R        | 3,783.16 | 3891/3884 | 628/1,329.96 | 228/4,511.00     | 706/1,444.92 | 0/0.00  | 166 | 82 | 1541 | 242      | 3720 | 4.60 | 95.40       | 5.7 |
| - shift       | 3,783.16 | -         | -            | 228/4,511.00     | 706/1,444.92 | -       | -   | -  | -    | -        | -    | -    | -           | 0.0 |
| 1860.216.30   | 4,784.88 | 3852/3851 | 805/3,126.24 | 213/3,502,389.40 | 886/3,976.28 | 13/0.70 | 165 | 83 | 1535 | 222      | 3720 | 3.55 | 96.45       | 5.7 |
| - no R        | 4,818.00 | 3878/3871 | 810/3,278.36 | 198/4,407.28     | 903/4,119.60 | 0/0.00  | 165 | 84 | 1547 | 222      | 3720 | 4.25 | 95.75       | 5.7 |
| - shift       | 4,818.00 | -         | -            | 198/4,407.28     | 903/4,119.60 | -       | -   | -  | -    | -        | -    | -    | -           | 0.3 |

Table 5.60: 3 weeks instances solved with a combined forward and backward search and  $\rho = 75\%$   $P$ .

| 3 weeks (f)   |          |           |              |                  |              |         |          |    |      |             |      |      |       |     |
|---------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-------------|------|------|-------|-----|
| $\rho = 75\%$ |          |           |              |                  |              |         |          |    |      |             |      |      |       |     |
| Solution      |          |           |              |                  |              |         | Criteria |    |      | Performance |      |      |       |     |
| T             | #M       | Wait      | Early        | Late             | R            |         | 0        | 1  | 2    | 3           | LB   | GAP  | QUAL  | CPU |
| 1860.98.50.f  | 2,822.04 | 3878/3872 | 538/638.84   | 245/830.686.24   | 453/460.36   | 3/0.16  | 167      | 77 | 1519 | 258         | 3720 | 4.25 | 95.75 | 5.5 |
| - no R        | 2,825.24 | 3884/3875 | 538/638.84   | 242/4,783.24     | 456/462.56   | 0/0.00  | 167      | 77 | 1522 | 258         | 3720 | 4.41 | 95.59 | 5.4 |
| - shift       | 2,825.24 | -         | -            | 242/4,783.24     | 456/462.56   | -       | -        | -  | -    | -           | -    | -    | -     | 0.0 |
| 1860.126.40.f | 3,427.32 | 3901/3894 | 600/927.56   | 227/1,091,609.56 | 621/905.20   | 4/0.22  | 168      | 77 | 1553 | 247         | 3720 | 4.87 | 95.13 | 5.4 |
| - no R        | 3,433.56 | 3909/3899 | 601/930.12   | 223/3,592.56     | 626/910.84   | 0/0.00  | 168      | 78 | 1556 | 247         | 3720 | 5.08 | 94.92 | 5.6 |
| - shift       | 3,433.56 | -         | -            | 223/3,592.56     | 626/910.84   | -       | -        | -  | -    | -           | -    | -    | -     | 0.0 |
| 1860.161.30.f | 4,157.44 | 3909/3904 | 698/1,836.96 | 232/1,632,159.48 | 777/2,011.88 | 6/0.32  | 166      | 83 | 1541 | 265         | 3720 | 5.08 | 94.92 | 5.5 |
| - no R        | 4,171.12 | 3921/3913 | 699/1,859.76 | 226/4,149.48     | 785/2,049.72 | 0/0.00  | 166      | 83 | 1547 | 265         | 3720 | 5.40 | 94.60 | 5.6 |
| - shift       | 4,171.12 | -         | -            | 226/4,149.48     | 785/2,049.72 | -       | -        | -  | -    | -           | -    | -    | -     | 0.1 |
| 1860.128.50.f | 3,194.44 | 3902/3896 | 569/798.48   | 250/830,666.44   | 545/755.24   | 3/0.16  | 170      | 86 | 1531 | 258         | 3720 | 4.89 | 95.11 | 5.5 |
| - no R        | 3,198.60 | 3908/3899 | 569/798.48   | 247/4,757.92     | 548/757.60   | 0/0.00  | 170      | 86 | 1534 | 258         | 3720 | 5.05 | 94.95 | 5.7 |
| - shift       | 3,198.60 | -         | -            | 247/4,757.92     | 548/757.60   | -       | -        | -  | -    | -           | -    | -    | -     | 0.0 |
| 1860.160.40.f | 3,770.60 | 3879/3875 | 624/1,312.36 | 234/1,632,524.00 | 698/1,418.76 | 6/0.32  | 166      | 82 | 1535 | 242         | 3720 | 4.27 | 95.73 | 5.5 |
| - no R        | 3,783.16 | 3891/3884 | 628/1,329.96 | 228/4,511.00     | 706/1,444.92 | 0/0.00  | 166      | 82 | 1541 | 242         | 3720 | 4.60 | 95.40 | 5.6 |
| - shift       | 3,783.16 | -         | -            | 228/4,511.00     | 706/1,444.92 | -       | -        | -  | -    | -           | -    | -    | -     | 0.0 |
| 1860.216.30.f | 4,784.88 | 3852/3851 | 805/3,126.24 | 213/3,502,389.40 | 886/3,976.28 | 13/0.70 | 165      | 83 | 1535 | 222         | 3720 | 3.55 | 96.45 | 5.6 |
| - no R        | 4,818.00 | 3878/3871 | 810/3,278.36 | 198/4,407.28     | 903/4,119.60 | 0/0.00  | 165      | 84 | 1547 | 222         | 3720 | 4.25 | 95.75 | 5.8 |
| - shift       | 4,818.00 | -         | -            | 198/4,407.28     | 903/4,119.60 | -       | -        | -  | -    | -           | -    | -    | -     | 0.3 |

Table 5.61: 3 weeks instances solved with a strict forward search and  $\rho = 75\% P$ .

| 3 weeks        |          | Solution  |              |                  |              |         | Criteria |    |      |     | Performance |      |       |     |
|----------------|----------|-----------|--------------|------------------|--------------|---------|----------|----|------|-----|-------------|------|-------|-----|
| $\rho = 100\%$ | T        | #M        | Wait         | Early            | Late         | R       | 0        | 1  | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| 1860.98.50     | 2,819.40 | 3885/3878 | 550/682.36   | 237/830,215.56   | 453/461.32   | 3/0.16  | 166      | 81 | 1538 | 243 | 3720        | 4.44 | 95.56 | 5.5 |
| - no R         | 2,822.20 | 3891/3881 | 550/682.36   | 234/4,312.56     | 456/463.28   | 0/0.00  | 166      | 81 | 1541 | 243 | 3720        | 4.60 | 95.40 | 5.4 |
| - shift        | 2,822.20 | -         | -            | 234/4,312.56     | 456/463.28   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.126.40    | 3,415.24 | 3890/3884 | 611/963.72   | 236/1,107,784.20 | 607/940.16   | 4/0.22  | 163      | 78 | 1541 | 252 | 3720        | 4.57 | 95.43 | 5.5 |
| - no R         | 3,421.24 | 3898/3888 | 612/968.92   | 232/3,997.68     | 613/950.16   | 0/0.00  | 163      | 78 | 1545 | 252 | 3720        | 4.78 | 95.22 | 5.5 |
| - shift        | 3,421.24 | -         | -            | 232/3,997.68     | 613/950.16   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.161.30    | 4,157.60 | 3884/3879 | 699/1,758.20 | 229/1,632,286.28 | 755/1,715.84 | 6/0.32  | 167      | 76 | 1536 | 251 | 3720        | 4.41 | 95.59 | 5.5 |
| - no R         | 4,170.48 | 3896/3888 | 700/1,780.20 | 223/4,276.28     | 762/1,752.96 | 0/0.00  | 167      | 76 | 1542 | 251 | 3720        | 4.73 | 95.27 | 5.6 |
| - shift        | 4,170.48 | -         | -            | 223/4,276.28     | 762/1,752.96 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.128.50    | 3,182.84 | 3877/3871 | 555/729.72   | 246/830,130.44   | 548/713.44   | 3/0.16  | 168      | 80 | 1540 | 232 | 3720        | 4.22 | 95.78 | 5.4 |
| - no R         | 3,187.00 | 3883/3874 | 555/729.72   | 243/4,224.52     | 551/715.96   | 0/0.00  | 168      | 80 | 1543 | 232 | 3720        | 4.38 | 95.62 | 5.5 |
| - shift        | 3,187.00 | -         | -            | 243/4,224.52     | 551/715.96   | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.160.40    | 3,779.24 | 3895/3891 | 628/1,271.88 | 239/1,354,842.84 | 707/1,355.40 | 5/0.27  | 172      | 80 | 1542 | 246 | 3720        | 4.70 | 95.30 | 5.5 |
| - no R         | 3,789.80 | 3905/3899 | 630/1,281.44 | 234/4,708.92     | 713/1,371.88 | 0/0.00  | 172      | 80 | 1547 | 246 | 3720        | 4.97 | 95.03 | 5.6 |
| - shift        | 3,789.80 | -         | -            | 234/4,708.92     | 713/1,371.88 | -       | -        | -  | -    | -   | -           | -    | -     | 0.0 |
| 1860.216.30    | 4,784.88 | 3852/3851 | 805/3,126.24 | 213/3,502,389.40 | 886/3,976.28 | 13/0.70 | 165      | 83 | 1535 | 222 | 3720        | 3.55 | 96.45 | 5.7 |
| - no R         | 4,818.00 | 3878/3871 | 810/3,278.36 | 198/4,407.28     | 903/4,119.60 | 0/0.00  | 165      | 84 | 1547 | 222 | 3720        | 4.25 | 95.75 | 5.7 |
| - shift        | 4,818.00 | -         | -            | 198/4,407.28     | 903/4,119.60 | -       | -        | -  | -    | -   | -           | -    | -     | 0.3 |

Table 5.62: 3 weeks instances solved with a combined forward and backward search and  $\rho = 100\%$   $P$ .

| 3 weeks (f)    |          | Solution  |              |                  |              |         |     |    | Criteria |     |      |      | Performance |      |     |  |
|----------------|----------|-----------|--------------|------------------|--------------|---------|-----|----|----------|-----|------|------|-------------|------|-----|--|
|                |          | T         | #M           | Wait             | Early        | Late    | R   | 0  | 1        | 2   | 3    | LB   | GAP         | QUAL | CPU |  |
| $\rho = 100\%$ |          |           |              |                  |              |         |     |    |          |     |      |      |             |      |     |  |
| 1860.98.50.f   | 2,817.16 | 3883/3877 | 541/592.08   | 232/829,940.12   | 475/503.92   | 3/0.16  | 166 | 79 | 1534     | 247 | 3720 | 4.38 | 95.62       | 5.5  |     |  |
| - no R         | 2,820.52 | 3889/3880 | 541/592.08   | 229/4,034.36     | 478/506.04   | 0/0.00  | 166 | 79 | 1537     | 247 | 3720 | 4.54 | 95.46       | 5.4  |     |  |
| - shift        | 2,820.52 | -         | -            | 229/4,034.36     | 478/506.04   | -       | -   | -  | -        | -   | -    | -    | -           | 0.0  |     |  |
| 1860.126.40.f  | 3,415.32 | 3890/3884 | 609/954.04   | 232/1,107,773.16 | 609/939.00   | 4/0.22  | 163 | 78 | 1541     | 252 | 3720 | 4.57 | 95.43       | 5.5  |     |  |
| - no R         | 3,421.32 | 3898/3888 | 610/959.24   | 228/3,986.64     | 615/949.00   | 0/0.00  | 163 | 78 | 1545     | 252 | 3720 | 4.78 | 95.22       | 5.6  |     |  |
| - shift        | 3,421.32 | -         | -            | 228/3,986.64     | 615/949.00   | -       | -   | -  | -        | -   | -    | -    | -           | 0.0  |     |  |
| 1860.161.30.f  | 4,157.60 | 3884/3879 | 699/1,758.20 | 229/1,632,286.28 | 755/1,715.84 | 6/0.32  | 167 | 76 | 1536     | 251 | 3720 | 4.41 | 95.59       | 5.6  |     |  |
| - no R         | 4,170.48 | 3896/3888 | 700/1,780.20 | 223/4,276.28     | 762/1,752.96 | 0/0.00  | 167 | 76 | 1542     | 251 | 3720 | 4.73 | 95.27       | 5.6  |     |  |
| - shift        | 4,170.48 | -         | -            | 223/4,276.28     | 762/1,752.96 | -       | -   | -  | -        | -   | -    | -    | -           | 0.0  |     |  |
| 1860.128.50.f  | 3,182.36 | 3877/3871 | 554/719.48   | 242/830,004.28   | 547/692.80   | 3/0.16  | 168 | 80 | 1540     | 232 | 3720 | 4.22 | 95.78       | 5.6  |     |  |
| - no R         | 3,186.52 | 3883/3874 | 554/719.48   | 239/4,098.36     | 550/695.32   | 0/0.00  | 168 | 80 | 1543     | 232 | 3720 | 4.38 | 95.62       | 5.9  |     |  |
| - shift        | 3,186.52 | -         | -            | 239/4,098.36     | 550/695.32   | -       | -   | -  | -        | -   | -    | -    | -           | 0.0  |     |  |
| 1860.160.40.f  | 3,779.24 | 3895/3891 | 628/1,271.88 | 239/1,354,842.84 | 707/1,355.40 | 5/0.27  | 172 | 80 | 1542     | 246 | 3720 | 4.70 | 95.30       | 5.7  |     |  |
| - no R         | 3,789.80 | 3905/3899 | 630/1,281.44 | 234/4,708.92     | 713/1,371.88 | 0/0.00  | 172 | 80 | 1547     | 246 | 3720 | 4.97 | 95.03       | 6.1  |     |  |
| - shift        | 3,789.80 | -         | -            | 234/4,708.92     | 713/1,371.88 | -       | -   | -  | -        | -   | -    | -    | -           | 0.0  |     |  |
| 1860.216.30.f  | 4,784.88 | 3852/3851 | 805/3,126.24 | 213/3,502,389.40 | 886/3,976.28 | 13/0.70 | 165 | 83 | 1535     | 222 | 3720 | 3.55 | 96.45       | 5.7  |     |  |
| - no R         | 4,818.00 | 3878/3871 | 810/3,278.36 | 198/4,407.28     | 903/4,119.60 | 0/0.00  | 165 | 84 | 1547     | 222 | 3720 | 4.25 | 95.75       | 5.7  |     |  |
| - shift        | 4,818.00 | -         | -            | 198/4,407.28     | 903/4,119.60 | -       | -   | -  | -        | -   | -    | -    | -           | 0.3  |     |  |

Table 5.63: 3 weeks instances solved with a strict forward search and  $\rho = 100\%$   $P$ .

| 4 weeks<br>$\rho = 25\%$ |  | Solution |           |               |                  |               | Criteria |     |    |      | Performance |      |       |       |     |
|--------------------------|--|----------|-----------|---------------|------------------|---------------|----------|-----|----|------|-------------|------|-------|-------|-----|
|                          |  | T        | #M        | Wait          | Early            | Late          | R        | 0   | 1  | 2    | 3           | LB   | GAP   | QUAL  | CPU |
| 2245.98.50               |  | 3,602.84 | 5430/5415 | 719/1,155.56  | 327/2,212,944.36 | 648/1,113.04  | 8/0.36   | 234 | 75 | 2002 | 882         | 4490 | 20.94 | 79.06 | 8.1 |
| - no R                   |  | 3,601.08 | 5462/5443 | 730/1,191.32  | 312/9,212.56     | 667/1,178.44  | 0/0.00   | 239 | 74 | 2006 | 898         | 4490 | 21.65 | 78.35 | 8.2 |
| - shift                  |  | 3,601.08 | -         | -             | 312/9,212.56     | 667/1,178.44  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.126.40              |  | 4,244.20 | 5189/5179 | 823/1,713.84  | 309/1,920,467.76 | 805/1,817.84  | 7/0.31   | 217 | 83 | 1952 | 699         | 4490 | 15.57 | 84.43 | 8.2 |
| - no R                   |  | 4,254.52 | 5215/5198 | 824/1,736.00  | 300/8,921.96     | 817/1,870.04  | 0/0.00   | 217 | 87 | 1960 | 706         | 4490 | 16.15 | 83.85 | 8.2 |
| - shift                  |  | 4,254.52 | -         | -             | 300/8,921.96     | 817/1,870.04  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.161.30              |  | 5,093.60 | 5023/5016 | 905/2,924.96  | 298/2,726,149.96 | 968/3,292.88  | 10/0.45  | 210 | 91 | 1900 | 587         | 4490 | 11.87 | 88.13 | 8.2 |
| - no R                   |  | 5,129.28 | 5067/5052 | 890/2,707.80  | 294/8,722.12     | 990/3,191.36  | 0/0.00   | 213 | 90 | 1914 | 605         | 4490 | 12.85 | 87.15 | 8.2 |
| - shift                  |  | 5,129.28 | -         | -             | 294/8,722.12     | 990/3,191.36  | -        | -   | -  | -    | -           | -    | -     | -     | 0.2 |
| 2245.128.50              |  | 4,012.76 | 5259/5249 | 744/1,422.16  | 319/1,917,920.64 | 731/1,491.32  | 7/0.31   | 223 | 82 | 1963 | 753         | 4490 | 17.13 | 82.87 | 8.2 |
| - no R                   |  | 4,030.28 | 5287/5271 | 744/1,437.68  | 314/8,041.68     | 742/1,497.68  | 0/0.00   | 226 | 82 | 1975 | 759         | 4490 | 17.75 | 82.25 | 8.1 |
| - shift                  |  | 4,030.28 | -         | -             | 314/8,041.68     | 742/1,497.68  | -        | -   | -  | -    | -           | -    | -     | -     | 0.1 |
| 2245.160.40              |  | 4,687.08 | 5139/5130 | 825/2,384.80  | 328/1,920,165.40 | 894/2,510.60  | 7/0.31   | 222 | 82 | 1921 | 676         | 4490 | 14.45 | 85.55 | 8.2 |
| - no R                   |  | 4,682.28 | 5127/5112 | 833/2,078.24  | 304/8,045.60     | 901/2,209.92  | 0/0.00   | 217 | 88 | 1923 | 654         | 4490 | 14.19 | 85.81 | 8.2 |
| - shift                  |  | 4,682.28 | -         | -             | 304/8,045.60     | 901/2,209.92  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.216.30              |  | 5,850.16 | 4839/4832 | 985/4,172.48  | 277/4,365,846.76 | 1121/4,828.52 | 16/0.71  | 208 | 84 | 1865 | 453         | 4490 | 7.77  | 92.23 | 8.3 |
| - no R                   |  | 5,892.56 | 4861/4848 | 1010/4,296.92 | 258/5,631.48     | 1123/4,853.36 | 0/0.00   | 203 | 87 | 1895 | 431         | 4490 | 8.26  | 91.74 | 8.2 |
| - shift                  |  | 5,892.56 | -         | -             | 258/5,631.48     | 1123/4,853.36 | -        | -   | -  | -    | -           | -    | -     | -     | 0.3 |

Table 5.64: 4 weeks instances solved with a combined forward and backward search and  $\rho = 25\%$   $P$ .

| 4 weeks (f)   |               | Solution |           |               |                  |               | Criteria |     |    |      | Performance |      |       |       |     |
|---------------|---------------|----------|-----------|---------------|------------------|---------------|----------|-----|----|------|-------------|------|-------|-------|-----|
|               |               | T        | #M        | Wait          | Early            | Late          | R        | 0   | 1  | 2    | 3           | LB   | GAP   | QUAL  | CPU |
| $\rho = 25\%$ | 2245.98.50.f  | 3,602.68 | 5450/5435 | 722/1,118.72  | 322/1,412,370.00 | 641/1,049.88  | 5/0.22   | 216 | 72 | 2019 | 903         | 4490 | 21.38 | 78.62 | 8.1 |
|               | - no R        | 3,580.68 | 5406/5385 | 720/1,111.64  | 344/1,011.12     | 630/1,040.64  | 0/0.00   | 235 | 75 | 1980 | 871         | 4490 | 20.40 | 79.60 | 8.2 |
|               | - shift       | 3,580.68 | -         | -             | 344/1,011.12     | 630/1,040.64  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.126.40.f | 2245.126.40.f | 4,244.68 | 5189/5179 | 819/1,703.84  | 309/1,920,463.48 | 806/1,859.24  | 7/0.31   | 217 | 84 | 1951 | 699         | 4490 | 15.57 | 84.43 | 8.4 |
|               | - no R        | 4,255.00 | 5215/5198 | 820/1,726.00  | 300/8,917.68     | 818/1,911.44  | 0/0.00   | 217 | 88 | 1959 | 706         | 4490 | 16.15 | 83.85 | 8.1 |
|               | - shift       | 4,255.00 | -         | -             | 300/8,917.68     | 818/1,911.44  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.161.30.f | 2245.161.30.f | 5,093.60 | 5023/5016 | 905/2,924.96  | 298/2,726,149.96 | 968/3,294.16  | 10/0.45  | 210 | 91 | 1900 | 587         | 4490 | 11.87 | 88.13 | 8.3 |
|               | - no R        | 5,129.04 | 5067/5052 | 889/2,703.40  | 294/8,722.12     | 991/3,200.16  | 0/0.00   | 213 | 90 | 1914 | 605         | 4490 | 12.85 | 87.15 | 8.2 |
|               | - shift       | 5,129.04 | -         | -             | 294/8,722.12     | 991/3,200.16  | -        | -   | -  | -    | -           | -    | -     | -     | 0.2 |
| 2245.128.50.f | 2245.128.50.f | 4,012.28 | 5259/5249 | 742/1,416.92  | 319/1,917,925.72 | 730/1,483.76  | 7/0.31   | 223 | 82 | 1963 | 753         | 4490 | 17.13 | 82.87 | 8.3 |
|               | - no R        | 4,029.80 | 5287/5271 | 742/1,432.44  | 314/8,046.76     | 741/1,490.12  | 0/0.00   | 226 | 82 | 1975 | 759         | 4490 | 17.75 | 82.25 | 8.2 |
|               | - shift       | 4,029.80 | -         | -             | 314/8,046.76     | 741/1,490.12  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.160.40.f | 2245.160.40.f | 4,687.08 | 5139/5130 | 825/2,384.80  | 328/1,920,165.40 | 894/2,510.60  | 7/0.31   | 222 | 82 | 1921 | 676         | 4490 | 14.45 | 85.55 | 9.0 |
|               | - no R        | 4,682.28 | 5127/5112 | 833/2,078.24  | 304/8,045.60     | 901/2,209.92  | 0/0.00   | 217 | 88 | 1923 | 654         | 4490 | 14.19 | 85.81 | 8.3 |
|               | - shift       | 4,682.28 | -         | -             | 304/8,045.60     | 901/2,209.92  | -        | -   | -  | -    | -           | -    | -     | -     | 0.0 |
| 2245.216.30.f | 2245.216.30.f | 5,850.16 | 4839/4832 | 985/4,172.48  | 277/4,365,846.76 | 1121/4,828.52 | 16/0.71  | 208 | 84 | 1865 | 453         | 4490 | 7.77  | 92.23 | 8.3 |
|               | - no R        | 5,892.56 | 4861/4848 | 1010/4,296.92 | 258/5,631.48     | 1123/4,853.36 | 0/0.00   | 203 | 87 | 1895 | 431         | 4490 | 8.26  | 91.74 | 8.2 |
|               | - shift       | 5,892.56 | -         | -             | 258/5,631.48     | 1123/4,853.36 | -        | -   | -  | -    | -           | -    | -     | -     | 0.3 |

Table 5.65: 4 weeks instances solved with a strict forward search and  $\rho = 25\%$   $P$ .

| 4 weeks<br>$\rho = 33\%$ |          | Solution  |              |                  |               |         | Criteria |     |      |     | Performance |       |       |      |
|--------------------------|----------|-----------|--------------|------------------|---------------|---------|----------|-----|------|-----|-------------|-------|-------|------|
|                          |          | T         | #M           | Wait             | Early         | Late    | R        | 0   | 1    | 2   | 3           | LB    | GAP   | QUAL |
| 2245.98.50               | 3,521.08 | 5236/5227 | 695/929.48   | 306/1,673,453.16 | 642/1,021.32  | 6/0.27  | 215      | 79  | 1968 | 735 | 4490        | 16.61 | 83.39 | 8.2  |
| - no R                   | 3,529.56 | 5231/5216 | 702/943.60   | 307/8,190.84     | 627/993.44    | 0/0.00  | 213      | 85  | 1971 | 717 | 4490        | 16.50 | 83.50 | 8.2  |
| - shift                  | 3,529.56 | -         | -            | 307/8,190.84     | 627/993.44    | -       | -        | -   | -    | -   | -           | -     | -     | 0.0  |
| 2245.126.40              | 4,201.64 | 5039/5029 | 745/1,413.64 | 320/1,935,314.76 | 759/1,417.76  | 7/0.31  | 215      | 79  | 1910 | 597 | 4490        | 12.23 | 87.77 | 8.3  |
| - no R                   | 4,214.36 | 5058/5043 | 750/1,424.28 | 311/8,099.80     | 772/1,442.48  | 0/0.00  | 214      | 79  | 1918 | 602 | 4490        | 12.65 | 87.35 | 8.1  |
| - shift                  | 4,214.36 | -         | -            | 311/8,099.80     | 772/1,442.48  | -       | -        | -   | -    | -   | -           | -     | -     | 0.0  |
| 2245.161.30              | 5,043.52 | 4791/4785 | 859/2,264.16 | 285/2,755,182.08 | 963/2,703.72  | 10/0.45 | 206      | 85  | 1855 | 410 | 4490        | 6.70  | 93.30 | 8.5  |
| - no R                   | 5,064.08 | 4811/4798 | 861/2,287.52 | 275/6,101.24     | 973/2,743.04  | 0/0.00  | 205      | 86  | 1865 | 410 | 4490        | 7.15  | 92.85 | 8.3  |
| - shift                  | 5,064.08 | -         | -            | 275/6,101.24     | 973/2,743.04  | -       | -        | -   | -    | -   | -           | -     | -     | 0.1  |
| 2245.128.50              | 3,947.80 | 5065/5056 | 727/1,227.76 | 314/1,133,441.56 | 725/1,195.08  | 4/0.18  | 210      | 81  | 1918 | 615 | 4490        | 12.81 | 87.19 | 8.3  |
| - no R                   | 3,952.84 | 5075/5062 | 727/1,229.36 | 312/8,273.00     | 730/1,204.64  | 0/0.00  | 210      | 81  | 1921 | 618 | 4490        | 13.03 | 86.97 | 8.5  |
| - shift                  | 3,952.84 | -         | -            | 312/8,273.00     | 730/1,204.64  | -       | -        | -   | -    | -   | -           | -     | -     | 0.0  |
| 2245.160.40              | 4,606.12 | 4862/4851 | 800/2,010.00 | 287/1,916,038.44 | 890/2,200.00  | 7/0.31  | 206      | 97  | 1881 | 440 | 4490        | 8.29  | 91.71 | 8.3  |
| - no R                   | 4,619.48 | 4876/4861 | 801/2,021.76 | 280/6,092.72     | 899/2,225.40  | 0/0.00  | 206      | 98  | 1887 | 440 | 4490        | 8.60  | 91.40 | 8.3  |
| - shift                  | 4,619.48 | -         | -            | 280/6,092.72     | 899/2,225.40  | -       | -        | -   | -    | -   | -           | -     | -     | 0.1  |
| 2245.216.30              | 5,817.04 | 4693/4685 | 992/3,754.08 | 302/3,292,412.16 | 1071/4,410.44 | 12/0.53 | 204      | 108 | 1822 | 326 | 4490        | 4.52  | 95.48 | 8.4  |
| - no R                   | 5,841.28 | 4713/4699 | 993/3,812.08 | 289/6,483.16     | 1086/4,492.64 | 0/0.00  | 204      | 108 | 1831 | 325 | 4490        | 4.97  | 95.03 | 8.2  |
| - shift                  | 5,841.28 | -         | -            | 289/6,483.16     | 1086/4,492.64 | -       | -        | -   | -    | -   | -           | -     | -     | 0.3  |

Table 5.66: 4 weeks instances solved with a combined forward and backward search and  $\rho = 33\%$   $P$ .

| 4 weeks (f)   |          |           |              |                  |               |         |          |     |      |             |      |       |       |     |
|---------------|----------|-----------|--------------|------------------|---------------|---------|----------|-----|------|-------------|------|-------|-------|-----|
| $\rho = 33\%$ |          |           |              |                  |               |         |          |     |      |             |      |       |       |     |
| Solution      |          |           |              |                  |               |         | Criteria |     |      | Performance |      |       |       |     |
|               | T        | #M        | Wait         | Early            | Late          | R       | 0        | 1   | 2    | 3           | LB   | GAP   | QUAL  | CPU |
| 2245.98.50.f  | 3,532.36 | 5186/5178 | 694/939.72   | 316/1,943,175.48 | 608/877.36    | 7/0.31  | 224      | 91  | 1941 | 692         | 4490 | 15.50 | 84.50 | 8.2 |
| - no R        | 3,539.40 | 5212/5197 | 693/930.60   | 290/8,394.48     | 647/966.28    | 0/0.00  | 227      | 77  | 1964 | 699         | 4490 | 16.08 | 83.92 | 8.3 |
| - shift       | 3,539.40 | -         | -            | 290/8,394.48     | 647/966.28    | -       | -        | -   | -    | -           | -    | -     | -     | 0.0 |
| 2245.126.40.f | 4,201.56 | 5039/5029 | 745/1,412.44 | 320/1,935,301.64 | 758/1,404.80  | 7/0.31  | 215      | 79  | 1910 | 597         | 4490 | 12.23 | 87.77 | 8.2 |
| - no R        | 4,214.28 | 5058/5043 | 750/1,423.08 | 311/8,086.68     | 771/1,429.52  | 0/0.00  | 214      | 79  | 1918 | 602         | 4490 | 12.65 | 87.35 | 8.2 |
| - shift       | 4,214.28 | -         | -            | 311/8,086.68     | 771/1,429.52  | -       | -        | -   | -    | -           | -    | -     | -     | 0.0 |
| 2245.161.30.f | 5,043.52 | 4791/4785 | 859/2,264.16 | 285/2,755,182.08 | 963/2,703.72  | 10/0.45 | 206      | 85  | 1855 | 410         | 4490 | 6.70  | 93.30 | 8.4 |
| - no R        | 5,064.08 | 4811/4798 | 861/2,287.52 | 275/6,101.24     | 973/2,743.04  | 0/0.00  | 205      | 86  | 1865 | 410         | 4490 | 7.15  | 92.85 | 8.5 |
| - shift       | 5,064.08 | -         | -            | 275/6,101.24     | 973/2,743.04  | -       | -        | -   | -    | -           | -    | -     | -     | 0.2 |
| 2245.128.50.f | 3,948.20 | 5065/5056 | 726/1,220.20 | 314/1,133,443.88 | 726/1,192.84  | 4/0.18  | 210      | 81  | 1918 | 615         | 4490 | 12.81 | 87.19 | 8.3 |
| - no R        | 3,953.24 | 5075/5062 | 726/1,221.80 | 312/8,275.32     | 731/1,202.40  | 0/0.00  | 210      | 81  | 1921 | 618         | 4490 | 13.03 | 86.97 | 8.4 |
| - shift       | 3,953.24 | -         | -            | 312/8,275.32     | 731/1,202.40  | -       | -        | -   | -    | -           | -    | -     | -     | 0.0 |
| 2245.160.40.f | 4,606.12 | 4862/4851 | 800/2,010.00 | 287/1,916,038.44 | 890/2,200.00  | 7/0.31  | 206      | 97  | 1881 | 440         | 4490 | 8.29  | 91.71 | 8.3 |
| - no R        | 4,619.48 | 4876/4861 | 801/2,021.76 | 280/6,092.72     | 899/2,225.40  | 0/0.00  | 206      | 98  | 1887 | 440         | 4490 | 8.60  | 91.40 | 8.6 |
| - shift       | 4,619.48 | -         | -            | 280/6,092.72     | 899/2,225.40  | -       | -        | -   | -    | -           | -    | -     | -     | 0.1 |
| 2245.216.30.f | 5,817.04 | 4693/4685 | 992/3,754.08 | 302/3,292,412.16 | 1071/4,410.44 | 12/0.53 | 204      | 108 | 1822 | 326         | 4490 | 4.52  | 95.48 | 8.4 |
| - no R        | 5,841.28 | 4713/4699 | 993/3,812.08 | 289/6,483.16     | 1086/4,492.64 | 0/0.00  | 204      | 108 | 1831 | 325         | 4490 | 4.97  | 95.03 | 8.5 |
| - shift       | 5,841.28 | -         | -            | 289/6,483.16     | 1086/4,492.64 | -       | -        | -   | -    | -           | -    | -     | -     | 0.3 |

Table 5.67: 4 weeks instances solved with a strict forward search and  $\rho = 33\%$   $P$ .



| 4 weeks       |          |           |              | Solution         |               |         |  | Criteria |    |      |     | Performance |       |       |     |
|---------------|----------|-----------|--------------|------------------|---------------|---------|--|----------|----|------|-----|-------------|-------|-------|-----|
| $\rho = 50\%$ | T        | #M        | Wait         | Early            | Late          | R       |  | 0        | 1  | 2    | 3   | LB          | GAP   | QUAL  | CPU |
| 2245.98.50    | 3,477.72 | 4972/4958 | 663/819.20   | 315/1,394,627.24 | 604/778.76    | 5/0.22  |  | 200      | 73 | 1918 | 541 | 4490        | 10.73 | 89.27 | 8.2 |
| - no R        | 3,484.12 | 4985/4966 | 664/824.60   | 311/7,290.32     | 608/785.52    | 0/0.00  |  | 200      | 74 | 1922 | 544 | 4490        | 11.02 | 88.98 | 8.1 |
| - shift       | 3,484.12 | -         | -            | 311/7,290.32     | 608/785.52    | -       |  | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 2245.126.40   | 4,128.68 | 4670/4658 | 714/980.08   | 304/1,392,916.96 | 752/1,018.08  | 5/0.22  |  | 205      | 88 | 1836 | 301 | 4490        | 4.01  | 95.99 | 8.9 |
| - no R        | 4,136.28 | 4680/4664 | 715/983.36   | 299/5,580.04     | 758/1,025.20  | 0/0.00  |  | 205      | 89 | 1840 | 301 | 4490        | 4.23  | 95.77 | 8.3 |
| - shift       | 4,136.28 | -         | -            | 299/5,580.04     | 758/1,025.20  | -       |  | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 2245.161.30   | 5,009.92 | 4654/4645 | 828/1,908.84 | 279/1,932,284.04 | 955/2,310.40  | 7/0.31  |  | 200      | 98 | 1849 | 269 | 4490        | 3.65  | 96.35 | 8.3 |
| - no R        | 5,025.20 | 4668/4655 | 828/1,933.56 | 272/4,954.00     | 963/2,352.92  | 0/0.00  |  | 200      | 98 | 1856 | 269 | 4490        | 3.96  | 96.04 | 8.3 |
| - shift       | 5,025.20 | -         | -            | 272/4,954.00     | 963/2,352.92  | -       |  | -        | -  | -    | -   | -           | -     | -     | 0.1 |
| 2245.128.50   | 3,858.28 | 4707/4696 | 668/931.60   | 324/1,132,157.20 | 709/1,032.88  | 4/0.18  |  | 206      | 98 | 1832 | 330 | 4490        | 4.83  | 95.17 | 8.3 |
| - no R        | 3,864.28 | 4715/4700 | 668/931.60   | 320/6,934.28     | 713/1,036.24  | 0/0.00  |  | 206      | 98 | 1836 | 330 | 4490        | 5.01  | 94.99 | 8.3 |
| - shift       | 3,864.28 | -         | -            | 320/6,934.28     | 713/1,036.24  | -       |  | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 2245.160.40   | 4,537.16 | 4674/4665 | 755/1,393.16 | 287/1,654,105.60 | 868/1,608.84  | 6/0.27  |  | 202      | 91 | 1854 | 288 | 4490        | 4.10  | 95.90 | 8.2 |
| - no R        | 4,547.96 | 4686/4673 | 756/1,402.88 | 281/4,654.68     | 874/1,624.16  | 0/0.00  |  | 202      | 91 | 1860 | 288 | 4490        | 4.37  | 95.63 | 8.3 |
| - shift       | 4,547.96 | -         | -            | 281/4,654.68     | 874/1,624.16  | -       |  | -        | -  | -    | -   | -           | -     | -     | 0.0 |
| 2245.216.30   | 5,796.96 | 4644/4639 | 978/3,700.80 | 276/3,822,739.84 | 1088/4,637.40 | 14/0.62 |  | 200      | 79 | 1854 | 280 | 4490        | 3.43  | 96.57 | 8.3 |
| - no R        | 5,832.00 | 4672/4660 | 987/3,854.96 | 262/5,564.44     | 1103/4,757.08 | 0/0.00  |  | 200      | 80 | 1867 | 280 | 4490        | 4.05  | 95.95 | 8.5 |
| - shift       | 5,832.00 | -         | -            | 262/5,564.44     | 1103/4,757.08 | -       |  | -        | -  | -    | -   | -           | -     | -     | 0.3 |

Table 5.68: 4 weeks instances solved with a combined forward and backward search and  $\rho = 50\%$   $P$ .

| 4 weeks (f)   |          |           |              |                  |               |          |     |    |      | Criteria |             |       |       |     | Performance |       |       |     |
|---------------|----------|-----------|--------------|------------------|---------------|----------|-----|----|------|----------|-------------|-------|-------|-----|-------------|-------|-------|-----|
| $\rho = 50\%$ | Solution |           |              |                  |               | Criteria |     |    |      |          | Performance |       |       |     | Performance |       |       |     |
|               | T        | #M        | Wait         | Early            | Late          | R        | 0   | 1  | 2    | 3        | LB          | GAP   | QUAL  | CPU | LB          | GAP   | QUAL  | CPU |
| 2245.98.50.f  | 3,468.36 | 4931/4917 | 658/800.96   | 318/1,394,693.32 | 571/680.80    | 5/0.22   | 201 | 82 | 1904 | 504      | 4490        | 9.82  | 90.18 | 8.7 | 4490        | 9.82  | 90.18 | 8.7 |
| - no R        | 3,474.36 | 4941/4923 | 659/802.64   | 313/7,355.32     | 576/685.76    | 0/0.00   | 201 | 82 | 1909 | 504      | 4490        | 10.04 | 89.96 | 8.2 | 4490        | 10.04 | 89.96 | 8.2 |
| - shift       | 3,474.36 | -         | -            | 313/7,355.32     | 576/685.76    | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 | -           | -     | -     | 0.0 |
| 2245.126.40.f | 4,128.92 | 4670/4658 | 714/976.72   | 303/1,392,916.00 | 753/1,023.32  | 5/0.22   | 205 | 88 | 1836 | 301      | 4490        | 4.01  | 95.99 | 8.4 | 4490        | 4.01  | 95.99 | 8.4 |
| - no R        | 4,136.52 | 4680/4664 | 715/980.00   | 298/5,579.08     | 759/1,030.44  | 0/0.00   | 205 | 89 | 1840 | 301      | 4490        | 4.23  | 95.77 | 8.4 | 4490        | 4.23  | 95.77 | 8.4 |
| - shift       | 4,136.52 | -         | -            | 298/5,579.08     | 759/1,030.44  | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 | -           | -     | -     | 0.0 |
| 2245.161.30.f | 5,009.92 | 4654/4645 | 828/1,908.84 | 279/1,932,284.04 | 955/2,310.40  | 7/0.31   | 200 | 98 | 1849 | 269      | 4490        | 3.65  | 96.35 | 8.3 | 4490        | 3.65  | 96.35 | 8.3 |
| - no R        | 5,025.20 | 4668/4655 | 828/1,933.56 | 272/4,954.00     | 963/2,352.92  | 0/0.00   | 200 | 98 | 1856 | 269      | 4490        | 3.96  | 96.04 | 8.3 | 4490        | 3.96  | 96.04 | 8.3 |
| - shift       | 5,025.20 | -         | -            | 272/4,954.00     | 963/2,352.92  | -        | -   | -  | -    | -        | -           | -     | -     | 0.1 | -           | -     | -     | 0.1 |
| 2245.128.50.f | 3,858.12 | 4707/4696 | 668/931.04   | 323/1,132,141.92 | 710/1,026.80  | 4/0.18   | 206 | 98 | 1832 | 330      | 4490        | 4.83  | 95.17 | 8.5 | 4490        | 4.83  | 95.17 | 8.5 |
| - no R        | 3,864.12 | 4715/4700 | 668/931.04   | 319/6,919.00     | 714/1,030.16  | 0/0.00   | 206 | 98 | 1836 | 330      | 4490        | 5.01  | 94.99 | 8.5 | 4490        | 5.01  | 94.99 | 8.5 |
| - shift       | 3,864.12 | -         | -            | 319/6,919.00     | 714/1,030.16  | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 | -           | -     | -     | 0.0 |
| 2245.160.40.f | 4,537.16 | 4674/4665 | 755/1,393.16 | 287/1,654,105.60 | 868/1,608.84  | 6/0.27   | 202 | 91 | 1854 | 288      | 4490        | 4.10  | 95.90 | 8.3 | 4490        | 4.10  | 95.90 | 8.3 |
| - no R        | 4,547.96 | 4686/4673 | 756/1,402.88 | 281/4,654.68     | 874/1,624.16  | 0/0.00   | 202 | 91 | 1860 | 288      | 4490        | 4.37  | 95.63 | 8.3 | 4490        | 4.37  | 95.63 | 8.3 |
| - shift       | 4,547.96 | -         | -            | 281/4,654.68     | 874/1,624.16  | -        | -   | -  | -    | -        | -           | -     | -     | 0.0 | -           | -     | -     | 0.0 |
| 2245.216.30.f | 5,796.96 | 4644/4639 | 978/3,700.80 | 276/3,822,739.84 | 1088/4,637.40 | 14/0.62  | 200 | 79 | 1854 | 280      | 4490        | 3.43  | 96.57 | 8.4 | 4490        | 3.43  | 96.57 | 8.4 |
| - no R        | 5,832.00 | 4672/4660 | 987/3,854.96 | 262/5,564.44     | 1103/4,757.08 | 0/0.00   | 200 | 80 | 1867 | 280      | 4490        | 4.05  | 95.95 | 8.5 | 4490        | 4.05  | 95.95 | 8.5 |
| - shift       | 5,832.00 | -         | -            | 262/5,564.44     | 1103/4,757.08 | -        | -   | -  | -    | -        | -           | -     | -     | 0.3 | -           | -     | -     | 0.3 |

Table 5.69: 4 weeks instances solved with a strict forward search and  $\rho = 50\%$   $P$ .

| 4 weeks       |          | Solution  |              |                  |               |         |     | Criteria |      |     |      | Performance |       |      |     |
|---------------|----------|-----------|--------------|------------------|---------------|---------|-----|----------|------|-----|------|-------------|-------|------|-----|
|               |          | T         | #M           | Wait             | Early         | Late    | R   | 0        | 1    | 2   | 3    | LB          | GAP   | QUAL | CPU |
| $\rho = 67\%$ |          |           |              |                  |               |         |     |          |      |     |      |             |       |      |     |
| 2245.98.50    | 3,411.24 | 4717/4707 | 655/721.52   | 293/1,130,505.20 | 594/670.60    | 4/0.18  | 209 | 87       | 1861 | 319 | 4490 | 5.06        | 94.94 | 8.4  |     |
| - no R        | 3,415.64 | 4725/4711 | 655/721.52   | 289/5,274.88     | 598/673.64    | 0/0.00  | 209 | 87       | 1865 | 319 | 4490 | 5.23        | 94.77 | 8.2  |     |
| - shift       | 3,415.64 | -         | -            | 289/5,274.88     | 598/673.64    | -       | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 2245.126.40   | 4,142.20 | 4680/4669 | 711/1,027.56 | 290/1,130,195.44 | 766/1,087.00  | 4/0.18  | 204 | 91       | 1842 | 302 | 4490 | 4.23        | 95.77 | 8.3  |     |
| - no R        | 4,148.28 | 4688/4673 | 711/1,027.56 | 286/4,964.56     | 770/1,090.60  | 0/0.00  | 204 | 91       | 1846 | 302 | 4490 | 4.41        | 95.59 | 8.4  |     |
| - shift       | 4,148.28 | -         | -            | 286/4,964.56     | 770/1,090.60  | -       | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 2245.161.30   | 5,019.28 | 4687/4677 | 837/2,100.00 | 289/1,932,176.88 | 947/2,381.44  | 7/0.31  | 200 | 101      | 1847 | 301 | 4490 | 4.39        | 95.61 | 8.5  |     |
| - no R        | 5,034.96 | 4701/4687 | 838/2,122.80 | 282/4,849.88     | 956/2,420.32  | 0/0.00  | 200 | 101      | 1854 | 301 | 4490 | 4.70        | 95.30 | 8.3  |     |
| - shift       | 5,034.96 | -         | -            | 282/4,849.88     | 956/2,420.32  | -       | -   | -        | -    | -   | -    | -           | -     | 0.1  |     |
| 2245.128.50   | 3,829.16 | 4682/4671 | 667/879.20   | 292/1,129,796.00 | 693/886.68    | 4/0.18  | 199 | 89       | 1860 | 293 | 4490 | 4.28        | 95.72 | 8.3  |     |
| - no R        | 3,834.92 | 4690/4675 | 667/879.20   | 288/4,576.00     | 697/889.88    | 0/0.00  | 199 | 89       | 1864 | 293 | 4490 | 4.45        | 95.55 | 8.3  |     |
| - shift       | 3,834.92 | -         | -            | 288/4,576.00     | 697/889.88    | -       | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 2245.160.40   | 4,547.48 | 4662/4653 | 754/1,503.04 | 286/1,932,440.72 | 856/1,676.84  | 7/0.31  | 199 | 96       | 1850 | 279 | 4490 | 3.83        | 96.17 | 8.4  |     |
| - no R        | 4,561.72 | 4676/4663 | 758/1,520.64 | 279/5,110.72     | 865/1,703.84  | 0/0.00  | 199 | 96       | 1857 | 279 | 4490 | 4.14        | 95.86 | 8.4  |     |
| - shift       | 4,561.72 | -         | -            | 279/5,110.72     | 865/1,703.84  | -       | -   | -        | -    | -   | -    | -           | -     | 0.0  |     |
| 2245.216.30   | 5,770.16 | 4626/4621 | 968/3,676.16 | 257/4,094,325.88 | 1096/4,635.60 | 15/0.67 | 203 | 98       | 1843 | 252 | 4490 | 3.03        | 96.97 | 8.5  |     |
| - no R        | 5,808.16 | 4656/4644 | 976/3,838.96 | 240/4,955.76     | 1115/4,791.36 | 0/0.00  | 203 | 99       | 1857 | 252 | 4490 | 3.70        | 96.30 | 8.8  |     |
| - shift       | 5,808.16 | -         | -            | 240/4,955.76     | 1115/4,791.36 | -       | -   | -        | -    | -   | -    | -           | -     | 0.3  |     |

Table 5.70: 4 weeks instances solved with a combined forward and backward search and  $\rho = 67\%$   $P$ .

| 4 weeks (f)   |          |           |              |                  |               |         |          |     |      |     |             |      |       |     |
|---------------|----------|-----------|--------------|------------------|---------------|---------|----------|-----|------|-----|-------------|------|-------|-----|
| $\rho = 67\%$ |          |           |              |                  |               |         |          |     |      |     |             |      |       |     |
| Solution      |          |           |              |                  |               |         | Criteria |     |      |     | Performance |      |       |     |
|               | T        | #M        | Wait         | Early            | Late          | R       | 0        | 1   | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| 2245.98.50.f  | 3,400.52 | 4686/4675 | 648/677.68   | 296/1,131,184.24 | 579/556.08    | 4/0.18  | 207      | 88  | 1838 | 312 | 4490        | 4.37 | 95.63 | 8.2 |
| - no R        | 3,405.32 | 4694/4679 | 648/677.68   | 292/5,961.40     | 583/559.12    | 0/0.00  | 207      | 88  | 1842 | 312 | 4490        | 4.54 | 95.46 | 8.2 |
| - shift       | 3,405.32 | -         | -            | 292/5,961.40     | 583/559.12    | -       | -        | -   | -    | -   | -           | -    | -     | 0.0 |
| 2245.126.40.f | 4,137.56 | 4682/4670 | 716/1,046.48 | 285/1,129,768.76 | 763/1,091.44  | 4/0.18  | 204      | 94  | 1848 | 295 | 4490        | 4.28 | 95.72 | 8.5 |
| - no R        | 4,143.40 | 4690/4674 | 716/1,046.48 | 281/4,538.12     | 767/1,094.80  | 0/0.00  | 204      | 94  | 1852 | 295 | 4490        | 4.45 | 95.55 | 8.2 |
| - shift       | 4,143.40 | -         | -            | 281/4,538.12     | 767/1,094.80  | -       | -        | -   | -    | -   | -           | -    | -     | 0.0 |
| 2245.161.30.f | 5,019.28 | 4687/4677 | 837/2,100.00 | 289/1,932,176.88 | 947/2,381.44  | 7/0.31  | 200      | 101 | 1847 | 301 | 4490        | 4.39 | 95.61 | 8.6 |
| - no R        | 5,034.96 | 4701/4687 | 838/2,122.80 | 282/4,849.88     | 956/2,420.32  | 0/0.00  | 200      | 101 | 1854 | 301 | 4490        | 4.70 | 95.30 | 8.4 |
| - shift       | 5,034.96 | -         | -            | 282/4,849.88     | 956/2,420.32  | -       | -        | -   | -    | -   | -           | -    | -     | 0.1 |
| 2245.128.50.f | 3,829.40 | 4682/4671 | 665/871.96   | 286/1,129,774.28 | 694/883.76    | 4/0.18  | 199      | 89  | 1860 | 293 | 4490        | 4.28 | 95.72 | 8.5 |
| - no R        | 3,835.16 | 4690/4675 | 665/871.96   | 282/4,554.28     | 698/886.96    | 0/0.00  | 199      | 89  | 1864 | 293 | 4490        | 4.45 | 95.55 | 8.2 |
| - shift       | 3,835.16 | -         | -            | 282/4,554.28     | 698/886.96    | -       | -        | -   | -    | -   | -           | -    | -     | 0.0 |
| 2245.160.40.f | 4,547.48 | 4662/4653 | 754/1,503.04 | 286/1,932,440.72 | 856/1,676.84  | 7/0.31  | 199      | 96  | 1850 | 279 | 4490        | 3.83 | 96.17 | 8.7 |
| - no R        | 4,561.72 | 4676/4663 | 758/1,520.64 | 279/5,110.72     | 865/1,703.84  | 0/0.00  | 199      | 96  | 1857 | 279 | 4490        | 4.14 | 95.86 | 8.3 |
| - shift       | 4,561.72 | -         | -            | 279/5,110.72     | 865/1,703.84  | -       | -        | -   | -    | -   | -           | -    | -     | 0.0 |
| 2245.216.30.f | 5,770.16 | 4626/4621 | 968/3,676.16 | 257/4,094,325.88 | 1096/4,635.60 | 15/0.67 | 203      | 98  | 1843 | 252 | 4490        | 3.03 | 96.97 | 8.7 |
| - no R        | 5,808.16 | 4656/4644 | 976/3,838.96 | 240/4,955.76     | 1115/4,791.36 | 0/0.00  | 203      | 99  | 1857 | 252 | 4490        | 3.70 | 96.30 | 8.5 |
| - shift       | 5,808.16 | -         | -            | 240/4,955.76     | 1115/4,791.36 | -       | -        | -   | -    | -   | -           | -    | -     | 0.3 |

Table 5.71: 4 weeks instances solved with a strict forward search and  $\rho = 67\%$   $P$ .

| 4 weeks       |  | Solution |           |              |                  |               | Criteria |     |     |      | Performance |      |      |       |     |
|---------------|--|----------|-----------|--------------|------------------|---------------|----------|-----|-----|------|-------------|------|------|-------|-----|
|               |  | T        | #M        | Wait         | Early            | Late          | R        | 0   | 1   | 2    | 3           | LB   | GAP  | QUAL  | CPU |
| $\rho = 75\%$ |  |          |           |              |                  |               |          |     |     |      |             |      |      |       |     |
| 2245.98.50    |  | 3,418.60 | 4664/4653 | 662/763.68   | 278/1,392,028.60 | 564/572.00    | 5/0.22   | 203 | 90  | 1855 | 276         | 4490 | 3.88 | 96.12 | 8.2 |
| - no R        |  | 3,424.92 | 4674/4660 | 663/767.16   | 273/4,694.60     | 570/578.72    | 0/0.00   | 203 | 90  | 1860 | 276         | 4490 | 4.10 | 95.90 | 8.2 |
| - shift       |  | 3,424.92 | -         | -            | 273/4,694.60     | 570/578.72    | -        | -   | -   | -    | -           | -    | -    | -     | 0.0 |
| 2245.126.40   |  | 4,129.48 | 4687/4676 | 720/1,085.40 | 274/1,391,538.84 | 770/1,109.68  | 5/0.22   | 203 | 95  | 1867 | 282         | 4490 | 4.39 | 95.61 | 8.5 |
| - no R        |  | 4,137.48 | 4697/4682 | 721/1,087.96 | 269/4,204.84     | 776/1,116.32  | 0/0.00   | 203 | 96  | 1871 | 282         | 4490 | 4.61 | 95.39 | 8.3 |
| - shift       |  | 4,137.48 | -         | -            | 269/4,204.84     | 776/1,116.32  | -        | -   | -   | -    | -           | -    | -    | -     | 0.0 |
| 2245.161.30   |  | 5,019.28 | 4687/4677 | 837/2,100.00 | 289/1,932,176.88 | 947/2,381.44  | 7/0.31   | 200 | 101 | 1847 | 301         | 4490 | 4.39 | 95.61 | 8.5 |
| - no R        |  | 5,034.96 | 4701/4687 | 838/2,122.80 | 282/4,849.88     | 956/2,420.32  | 0/0.00   | 200 | 101 | 1854 | 301         | 4490 | 4.70 | 95.30 | 8.3 |
| - shift       |  | 5,034.96 | -         | -            | 282/4,849.88     | 956/2,420.32  | -        | -   | -   | -    | -           | -    | -    | -     | 0.1 |
| 2245.128.50   |  | 3,843.40 | 4678/4667 | 684/935.24   | 303/1,130,591.44 | 683/931.92    | 4/0.18   | 205 | 101 | 1840 | 291         | 4490 | 4.19 | 95.81 | 8.5 |
| - no R        |  | 3,849.08 | 4686/4671 | 684/935.24   | 299/5,365.92     | 687/935.12    | 0/0.00   | 205 | 101 | 1844 | 291         | 4490 | 4.37 | 95.63 | 8.4 |
| - shift       |  | 3,849.08 | -         | -            | 299/5,365.92     | 687/935.12    | -        | -   | -   | -    | -           | -    | -    | -     | 0.0 |
| 2245.160.40   |  | 4,547.64 | 4662/4653 | 754/1,502.24 | 286/1,932,440.72 | 855/1,676.20  | 7/0.31   | 199 | 96  | 1850 | 279         | 4490 | 3.83 | 96.17 | 8.8 |
| - no R        |  | 4,561.88 | 4676/4663 | 758/1,519.84 | 279/5,110.72     | 864/1,703.20  | 0/0.00   | 199 | 96  | 1857 | 279         | 4490 | 4.14 | 95.86 | 8.4 |
| - shift       |  | 4,561.88 | -         | -            | 279/5,110.72     | 864/1,703.20  | -        | -   | -   | -    | -           | -    | -    | -     | 0.0 |
| 2245.216.30   |  | 5,770.16 | 4626/4621 | 968/3,676.16 | 257/4,094,325.88 | 1096/4,635.60 | 15/0.67  | 203 | 98  | 1843 | 252         | 4490 | 3.03 | 96.97 | 8.8 |
| - no R        |  | 5,808.16 | 4656/4644 | 976/3,838.96 | 240/4,955.76     | 1115/4,791.36 | 0/0.00   | 203 | 99  | 1857 | 252         | 4490 | 3.70 | 96.30 | 8.5 |
| - shift       |  | 5,808.16 | -         | -            | 240/4,955.76     | 1115/4,791.36 | -        | -   | -   | -    | -           | -    | -    | -     | 0.3 |

Table 5.72: 4 weeks instances solved with a combined forward and backward search and  $\rho = 75\%$   $P$ .

| 4 weeks (f)   |          |           |              |                  |               |         |     |     |      | Criteria |      |      |       |      | Performance |  |  |  |
|---------------|----------|-----------|--------------|------------------|---------------|---------|-----|-----|------|----------|------|------|-------|------|-------------|--|--|--|
| Solution      |          |           |              |                  |               |         |     |     |      | Criteria |      |      |       |      | Performance |  |  |  |
| $\rho = 75\%$ | T        | #M        | Wait         | Early            | Late          | R       | 0   | 1   | 2    | 3        | LB   | GAP  | QUAL  | CPU  |             |  |  |  |
| 2245.98.50.f  | 3,403.48 | 4659/4648 | 651/745.84   | 290/1,130,626.00 | 561/555.36    | 4/0.18  | 204 | 91  | 1835 | 288      | 4490 | 3.76 | 96.24 | 8.1  |             |  |  |  |
| - no R        | 3,408.04 | 4667/4652 | 651/745.84   | 286/5,406.00     | 565/558.32    | 0/0.00  | 204 | 91  | 1839 | 288      | 4490 | 3.94 | 96.06 | 8.4  |             |  |  |  |
| - shift       | 3,408.04 | -         | -            | 286/5,406.00     | 565/558.32    | -       | -   | -   | -    | -        | -    | -    | -     | 0.0  |             |  |  |  |
| 2245.126.40.f | 4,129.32 | 4687/4675 | 718/1,072.28 | 273/1,391,534.40 | 770/1,101.80  | 5/0.22  | 203 | 95  | 1867 | 282      | 4490 | 4.39 | 95.61 | 8.5  |             |  |  |  |
| - no R        | 4,137.32 | 4697/4681 | 719/1,074.84 | 268/4,200.40     | 776/1,108.44  | 0/0.00  | 203 | 96  | 1871 | 282      | 4490 | 4.61 | 95.39 | 8.5  |             |  |  |  |
| - shift       | 4,137.32 | -         | -            | 268/4,200.40     | 776/1,108.44  | -       | -   | -   | -    | -        | -    | -    | -     | 0.0  |             |  |  |  |
| 2245.161.30.f | 5,019.28 | 4687/4677 | 837/2,100.00 | 289/1,932,176.88 | 947/2,381.44  | 7/0.31  | 200 | 101 | 1847 | 301      | 4490 | 4.39 | 95.61 | 8.5  |             |  |  |  |
| - no R        | 5,034.96 | 4701/4687 | 838/2,122.80 | 282/4,849.88     | 956/2,420.32  | 0/0.00  | 200 | 101 | 1854 | 301      | 4490 | 4.70 | 95.30 | 8.5  |             |  |  |  |
| - shift       | 5,034.96 | -         | -            | 282/4,849.88     | 956/2,420.32  | -       | -   | -   | -    | -        | -    | -    | -     | 0.1  |             |  |  |  |
| 2245.128.50.f | 3,843.64 | 4678/4667 | 684/929.24   | 301/1,130,584.88 | 682/915.72    | 4/0.18  | 205 | 101 | 1840 | 291      | 4490 | 4.19 | 95.81 | 8.2  |             |  |  |  |
| - no R        | 3,849.32 | 4686/4671 | 684/929.24   | 297/5,359.36     | 686/918.92    | 0/0.00  | 205 | 101 | 1844 | 291      | 4490 | 4.37 | 95.63 | 8.4  |             |  |  |  |
| - shift       | 3,849.32 | -         | -            | 297/5,359.36     | 686/918.92    | -       | -   | -   | -    | -        | -    | -    | -     | 0.0  |             |  |  |  |
| 2245.160.40.f | 4,547.64 | 4662/4653 | 754/1,502.24 | 286/1,932,440.72 | 855/1,676.20  | 7/0.31  | 199 | 96  | 1850 | 279      | 4490 | 3.83 | 96.17 | 8.7  |             |  |  |  |
| - no R        | 4,561.88 | 4676/4663 | 758/1,519.84 | 279/5,110.72     | 864/1,703.20  | 0/0.00  | 199 | 96  | 1857 | 279      | 4490 | 4.14 | 95.86 | 8.5  |             |  |  |  |
| - shift       | 4,561.88 | -         | -            | 279/5,110.72     | 864/1,703.20  | -       | -   | -   | -    | -        | -    | -    | -     | 0.0  |             |  |  |  |
| 2245.216.30.f | 5,770.16 | 4626/4621 | 968/3,676.16 | 257/4,094,325.88 | 1096/4,635.60 | 15/0.67 | 203 | 98  | 1843 | 252      | 4490 | 3.03 | 96.97 | 8.7  |             |  |  |  |
| - no R        | 5,808.16 | 4656/4644 | 976/3,838.96 | 240/4,955.76     | 1115/4,791.36 | 0/0.00  | 203 | 99  | 1857 | 252      | 4490 | 3.70 | 96.30 | 11.6 |             |  |  |  |
| - shift       | 5,808.16 | -         | -            | 240/4,955.76     | 1115/4,791.36 | -       | -   | -   | -    | -        | -    | -    | -     | 0.4  |             |  |  |  |

Table 5.73: 4 weeks instances solved with a strict forward search and  $\rho = 75\%$   $P$ .

| 4 weeks        |          | Solution  |              |                  |               |         |     | Criteria |      |     | Performance |      |       |     |
|----------------|----------|-----------|--------------|------------------|---------------|---------|-----|----------|------|-----|-------------|------|-------|-----|
| $\rho = 100\%$ | T        | #M        | Wait         | Early            | Late          | R       | 0   | 1        | 2    | 3   | LB          | GAP  | QUAL  | CPU |
| 2245.98.50     | 3,401.72 | 4663/4651 | 663/776.44   | 285/1,130,032.68 | 573/578.80    | 4/0.18  | 198 | 93       | 1852 | 279 | 4490        | 3.85 | 96.15 | 8.3 |
| - no R         | 3,405.88 | 4671/4655 | 663/776.44   | 281/4,812.68     | 577/581.52    | 0/0.00  | 198 | 93       | 1856 | 279 | 4490        | 4.03 | 95.97 | 8.3 |
| - shift        | 3,405.88 | -         | -            | 281/4,812.68     | 577/581.52    | -       | -   | -        | -    | -   | -           | -    | -     | 0.0 |
| 2245.126.40    | 4,122.92 | 4670/4659 | 739/1,130.36 | 290/1,408,012.24 | 759/1,174.16  | 5/0.22  | 198 | 92       | 1851 | 289 | 4490        | 4.01 | 95.99 | 8.7 |
| - no R         | 4,130.68 | 4680/4664 | 740/1,135.56 | 285/4,908.72     | 766/1,185.16  | 0/0.00  | 198 | 92       | 1856 | 289 | 4490        | 4.23 | 95.77 | 8.6 |
| - shift        | 4,130.68 | -         | -            | 285/4,908.72     | 766/1,185.16  | -       | -   | -        | -    | -   | -           | -    | -     | 0.0 |
| 2245.161.30    | 5,010.24 | 4671/4661 | 840/2,031.52 | 286/1,932,411.44 | 930/2,140.36  | 7/0.31  | 202 | 94       | 1848 | 289 | 4490        | 4.03 | 95.97 | 8.7 |
| - no R         | 5,025.12 | 4685/4671 | 841/2,053.52 | 279/5,084.44     | 938/2,178.52  | 0/0.00  | 202 | 94       | 1855 | 289 | 4490        | 4.34 | 95.66 | 8.9 |
| - shift        | 5,025.12 | -         | -            | 279/5,084.44     | 938/2,178.52  | -       | -   | -        | -    | -   | -           | -    | -     | 0.0 |
| 2245.128.50    | 3,822.84 | 4658/4647 | 671/875.52   | 299/1,130,328.72 | 678/860.52    | 4/0.18  | 200 | 98       | 1849 | 270 | 4490        | 3.74 | 96.26 | 8.5 |
| - no R         | 3,828.36 | 4666/4651 | 671/875.52   | 295/5,105.80     | 682/863.64    | 0/0.00  | 200 | 98       | 1853 | 270 | 4490        | 3.92 | 96.08 | 8.6 |
| - shift        | 3,828.36 | -         | -            | 295/5,105.80     | 682/863.64    | -       | -   | -        | -    | -   | -           | -    | -     | 0.0 |
| 2245.160.40    | 4,552.20 | 4672/4663 | 750/1,444.40 | 291/1,655,321.40 | 863/1,621.76  | 6/0.27  | 205 | 91       | 1858 | 279 | 4490        | 4.05 | 95.95 | 8.5 |
| - no R         | 4,564.60 | 4684/4672 | 752/1,453.96 | 285/5,870.48     | 870/1,639.24  | 0/0.00  | 205 | 91       | 1864 | 279 | 4490        | 4.32 | 95.68 | 8.8 |
| - shift        | 4,564.60 | -         | -            | 285/5,870.48     | 870/1,639.24  | -       | -   | -        | -    | -   | -           | -    | -     | 0.0 |
| 2245.216.30    | 5,770.16 | 4626/4621 | 968/3,676.16 | 257/4,094,325.88 | 1096/4,635.60 | 15/0.67 | 203 | 98       | 1843 | 252 | 4490        | 3.03 | 96.97 | 8.7 |
| - no R         | 5,808.16 | 4656/4644 | 976/3,838.96 | 240/4,955.76     | 1115/4,791.36 | 0/0.00  | 203 | 99       | 1857 | 252 | 4490        | 3.70 | 96.30 | 8.8 |
| - shift        | 5,808.16 | -         | -            | 240/4,955.76     | 1115/4,791.36 | -       | -   | -        | -    | -   | -           | -    | -     | 0.3 |

Table 5.74: 4 weeks instances solved with a combined forward and backward search and  $\rho = 100\%$   $P$ .

| 4 weeks (f)   |          |           |              |                  |               |         |     |    |      | Criteria |      |      |       | Performance |  |  |  |
|---------------|----------|-----------|--------------|------------------|---------------|---------|-----|----|------|----------|------|------|-------|-------------|--|--|--|
| Solution      |          |           |              |                  |               |         |     |    |      | 0        |      |      |       | LB          |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | 1        |      |      |       | GAP         |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | 2        |      |      |       | QUAL        |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | 3        |      |      |       | CPU         |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | R        |      |      |       |             |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | Late     |      |      |       |             |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | Early    |      |      |       |             |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | Wait     |      |      |       |             |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | #M       |      |      |       |             |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | T        |      |      |       |             |  |  |  |
|               |          |           |              |                  |               |         |     |    |      | ρ = 100% |      |      |       |             |  |  |  |
| 2245.98.50.f  | 3,398.68 | 4664/4653 | 653/692.20   | 277/1,129,898.04 | 585/590.08    | 4/0.18  | 202 | 97 | 1848 | 276      | 4490 | 3.88 | 96.12 | 8.3         |  |  |  |
| - no R        | 3,403.40 | 4672/4657 | 653/692.20   | 273/4,675.28     | 589/592.96    | 0/0.00  | 202 | 97 | 1852 | 276      | 4490 | 4.05 | 95.95 | 8.2         |  |  |  |
| - shift       | 3,403.40 | -         | -            | 273/4,675.28     | 589/592.96    | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |  |  |  |
| 2245.126.40.f | 4,123.00 | 4670/4659 | 737/1,120.68 | 286/1,408,003.20 | 761/1,173.00  | 5/0.22  | 198 | 92 | 1851 | 289      | 4490 | 4.01 | 95.99 | 8.5         |  |  |  |
| - no R        | 4,130.76 | 4680/4664 | 738/1,125.88 | 281/4,899.68     | 768/1,184.00  | 0/0.00  | 198 | 92 | 1856 | 289      | 4490 | 4.23 | 95.77 | 8.3         |  |  |  |
| - shift       | 4,130.76 | -         | -            | 281/4,899.68     | 768/1,184.00  | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |  |  |  |
| 2245.161.30.f | 5,010.24 | 4671/4661 | 840/2,031.52 | 286/1,932,411.44 | 930/2,140.36  | 7/0.31  | 202 | 94 | 1848 | 289      | 4490 | 4.03 | 95.97 | 8.6         |  |  |  |
| - no R        | 5,025.12 | 4685/4671 | 841/2,053.52 | 279/5,084.44     | 938/2,178.52  | 0/0.00  | 202 | 94 | 1855 | 289      | 4490 | 4.34 | 95.66 | 8.9         |  |  |  |
| - shift       | 5,025.12 | -         | -            | 279/5,084.44     | 938/2,178.52  | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |  |  |  |
| 2245.128.50.f | 3,822.36 | 4658/4647 | 670/865.28   | 296/1,130,318.24 | 677/839.88    | 4/0.18  | 200 | 98 | 1849 | 270      | 4490 | 3.74 | 96.26 | 8.3         |  |  |  |
| - no R        | 3,827.88 | 4666/4651 | 670/865.28   | 292/5,095.32     | 681/843.00    | 0/0.00  | 200 | 98 | 1853 | 270      | 4490 | 3.92 | 96.08 | 9.0         |  |  |  |
| - shift       | 3,827.88 | -         | -            | 292/5,095.32     | 681/843.00    | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |  |  |  |
| 2245.160.40.f | 4,552.20 | 4672/4663 | 750/1,444.40 | 291/1,655,321.40 | 863/1,621.76  | 6/0.27  | 205 | 91 | 1858 | 279      | 4490 | 4.05 | 95.95 | 9.1         |  |  |  |
| - no R        | 4,564.60 | 4684/4672 | 752/1,453.96 | 285/5,870.48     | 870/1,639.24  | 0/0.00  | 205 | 91 | 1864 | 279      | 4490 | 4.32 | 95.68 | 8.9         |  |  |  |
| - shift       | 4,564.60 | -         | -            | 285/5,870.48     | 870/1,639.24  | -       | -   | -  | -    | -        | -    | -    | -     | 0.0         |  |  |  |
| 2245.216.30.f | 5,770.16 | 4626/4621 | 968/3,676.16 | 257/4,094,325.88 | 1096/4,635.60 | 15/0.67 | 203 | 98 | 1843 | 252      | 4490 | 3.03 | 96.97 | 9.7         |  |  |  |
| - no R        | 5,808.16 | 4656/4644 | 976/3,838.96 | 240/4,955.76     | 1115/4,791.36 | 0/0.00  | 203 | 99 | 1857 | 252      | 4490 | 3.70 | 96.30 | 9.1         |  |  |  |
| - shift       | 5,808.16 | -         | -            | 240/4,955.76     | 1115/4,791.36 | -       | -   | -  | -    | -        | -    | -    | -     | 0.3         |  |  |  |

Table 5.75: 4 weeks instances solved with a strict forward search and  $\rho = 100 \% P$ .



## Discussion of the heuristic large-scale results

All 1 week problem instances are solved in less than 2.0 seconds and, as was the case in the small- and medium-scale test phases, run times for the shifting routine do not exceed 0.1 seconds except for a few instances with  $P = 216$ . The quality measured against the lower bound of the number of full moves varies between 81.19 % and 95.62 % which indicates a limited amount of container reshuffling in general. In the 856.128.50 instance, when not allowing rejections, a total of 2,034 full moves are performed, i.e. 322 containers are reshuffled, whereas the solutions with  $QUAL = 95.62$  %, occurring only in the 856.216.30 instances, imply only 75 reshuffles. There is a clear tendency of better quality when increasing  $P$  and, furthermore, when decreasing the  $R/B$  ratio. This may be due to less need for stacking and reshuffling when the load factor  $C/P$  is small and shorter moving distances when the block is oblong rather than closer to square. Moreover,  $QUAL$  increases the larger the search range  $\rho$  which is not surprising as searching among all available positions enables better choices and since this effort does not result in longer run times, a search range of 100 %  $P$  must be recommended.

The criteria statistics show that the number of moves performed by criterion 0 varies between 94 and 100, not changing by the different search strategies or block sizes. A similar result is seen for criterion 1, varying between 27 and 43 throughout all 1 week runs. The number of moves performed by criterion 2 varies between 685 and 745 and there is a tendency of decrease when increasing  $\rho$  which is also reflected in  $QUAL$  as discussed above. Criterion 3 accounts for 109 - 304 moves and there is a clear indication of smaller numbers when increasing  $\rho$  and  $P$ , but also when decreasing the  $R/B$  ratio, the explanation lying in the increased chances of finding good positions when searching among a large number of positions and when moving the containers over shorter distances, more or less in a straight line from arrival place to departure place. There is, for obvious reasons, a clear inverse proportional connection between the number of moves performed by criterion 3 and the quality.

The number of rejected containers does not exceed 6, corresponding to 0.70 % of  $C$ , in any of the instances. The largest number of rejections occurs in the 856.216.30 and 856.216.30.f instances and - with a few exceptions - the smallest number is found in the 856.128.50 and 856.128.50.f instances. Contrary to previous results, this indicates that better performance in the number of rejections is obtained by high  $R/B$  ratios and not too large number of positions whereas  $\rho$  does not seem to have any influence in this matter. Not surprising, the quality decreases slightly when preventing rejections as the initially rejected containers are forced to be positioned. The extra moves caused by the strict rule of positioning all containers are in general performed by criterion 2 and 3, the 856.216.30 and 856.216.30.f instances being particularly interesting as none of the extra moves are performed by criterion 3 which is, presumably, due to the large number of positions to choose between. Moreover, not allowing rejections seems to have a positive effect on the total time containers are early for departure in several cases.

The total waiting time is considerably longer in the 856.216.30 and 856.216.30.f instances, the longest accumulated time of 1,546 minutes occurring with  $\rho = 25$  % of  $P$ . The tendency is the same for the total amount of time containers are early or late

for departure. This indicates that even though fewer moves are required when a large number of positions is available, better results in terms of punctuality are obtained if the block - or the number of positions reserved for the 20' containers - is relatively small. In fact, the best results concerning few delays are found in the 856.98.50 and 856.98.50.f instances, otherwise representing the poorest outcome. It should be mentioned that if setting  $\mu = 0$ , naturally, no containers are early. Another interesting but discouraging result is that no improvement on the initial heuristic solution is made by the shifting routine in any of the 1 week problem instances. The explanation for this may be the crane velocity, equal to 1 in the small- and medium-scale test phases in order to provide transparent results that are easy to investigate and evaluate, but set to the realistic value of 25, corresponding to about 10 km/h, in the large-scale instances. This makes it possible to perform a large number of consecutive moves in short time but, in addition, it leaves no time intervals long enough for delayed moves to be shifted. Furthermore, the large number of containers, naturally, may create more time conflicts if shifting single moves. Therefore, it must be concluded that the improvement routine is not effective in its present form and, rather, other approaches to repair poor initial solutions should be investigated.

As well as for the 1 week subclass, run times are short, not exceeding 4.0 seconds, when solving the 2 weeks subclass instances and, again, running the shifting routine consumes only a fraction of a second. The quality in terms of the number of full moves varies between 79.54 % and 96.13 % with the same tendency towards higher quality the larger  $P$  and  $\rho$  and the smaller  $R/B$  showing in this subclass. As seen in the 1 week instances results, the number of moves performed by criterion 0 does not change much throughout the different sets of problem runs, varying between 126 and 140. Also, the 42 - 66 criterion 1 moves seems unaffected by changes in block layout and search strategy. Contrary to the 1 week subclass results, there is not a clear tendency of decreasing number of moves performed by criterion 2 when increasing  $\rho$ , the result varying from 1,100 to 1,214 across the different runs. On the other hand, the search range and block layout have a significant effect on the number of criterion 3 moves, varying from 172 in some of the instances with  $\rho \geq \frac{2}{3}P$ ,  $P = 216$ , and  $R/B = 30$  % to around 530 for instances with  $\rho = \frac{1}{4}P$  and  $R/B = 50$  %, indicating that the smallest number of reshuffles, proportional to a high quality, is reached with a large search range and a small number of rows compared with the number of bays, in line with the 1 week results.

The number of containers rejected by the basic heuristic algorithm does not exceed 1.0 % of  $C$  and, again, the largest number of rejections occurs in the 1356.216.30 and 1356.216.30.f instances, in particular when  $\rho$  equals  $\frac{1}{4}P$  where 11 containers are not positioned. However, the search range does not seem to influence the rejection rate in a systematic way whereas the  $R/B$  ratio has an effect: The larger the fewer rejections. Contrary to results for the 1 week subclass, none of the instances solve without rejecting containers when running the basic algorithm. Again, a small decrease in QUAL follows from not accepting rejections, in general, the vast majority of the extra moves forced by the -no R runs being performed by criterion 2. Furthermore, the tendency of a positive influence on the extend to which containers are early, seen in the 1 week subclass, shows

in the 2 weeks results as well.

In line with the 1 week subclass results, both accumulated waiting time and delays increase with larger values of  $P$  and smaller  $R/B$  ratios whereas the size of  $\rho$  is without importance, again indicating that if seeking a high degree of punctuality rather than few reshuffles, smaller and more square blocks are preferred. Furthermore, the failure of the shifting routine is repeated.

Run times for the 3 and 4 weeks instances are all under 10.0 seconds and the shifting routine finishes within a fraction of a second as well as for all previous results. The quality varies between 77.30 % and 96.97 % throughout the 3 and 4 weeks subclasses results, again instances with large values of  $\rho$  and  $P$  and small  $R/B$  ratios performing best in terms of minimizing the number of full moves. The number of criterion 0 and 1 moves are, again in general, unaffected by changes in settings for the runs. There is a faint tendency of a decreasing number of moves performed by criterion 2 when increasing  $\rho$  whereas it is more or less indifferent to the number of positions and the proportion of the block. The number of criterion 3 moves is significantly affected in a positive way when increasing the number of positions and the search range and decreasing the  $R/B$  ratio, resembling the results from the 1 week and 2 weeks subclasses.

As well as for previously discussed results, the rejection rate does not exceed 1.0 % of  $C$  in any of the 3 or 4 weeks instances, the worst performance in this respect again obtained in the instances with 216 positions and a  $R/B$  ratio of 30 % and the best results found in runs with  $R/B = 0.5$ . As well as for the 2 weeks subclass, no 3 or 4 weeks instances, solved by the basic heuristic algorithm, results in a rejection rate of 0, the extra moves required when not allowing rejections primarily being performed by criterion 2. In line with previous large-scale results, containers are much less early for departure when running the algorithm without rejections.

Finally, the same connection between large values of  $P$ , small values of  $R/B$ , and a great extent of waiting time at the arrival place and delays at the departure place is seen in the 3 and 4 weeks instances and, again, the shifting routine does not succeed in improving the solutions.

#### 5.2.4 Conclusive remarks on the heuristic results

Concluding, the results from the large-scale class problems point in two different directions for which reason the best choice of strategy depends on the objective which may vary depending on the port authority or terminal managers. If seeking a minimal number of moves and reshuffles, reflected in a small number of criterion 3 moves and a high quality in the presented results, a large number of positions should be reserved for the containers, the block should be far from square, e.g. three times as many bays as rows, and the full set of positions should be taken into consideration when positioning containers. On the other hand, if aiming at a high degree of punctuality, the opposite strategy should be applied, limiting the number of positions and keeping a rather large  $R/B$  ratio, whereas the choice of  $\rho$  is less important. In general, neither the combined forward and backward nor the strict forward search strategy outperforms the other, indicating that the search range rather than the direction may be of importance.

The heuristic approach to solving the CPP seems to be a good alternative to using the presented models and standard software. Run times are very short, regardless the size of the instances, high-quality solutions are obtained, and a wide range of different strategies can be applied, depending on the demanded results. Whereas the size of  $P$  and  $R/B$  has significant influence on the outcome and should be chosen with care,  $\rho$  can always be set to 100 % of  $P$  as this setting has a positive effect in some respects and no influence in others and large values does not increase run times so that this issue becomes problematic. The choice of search strategy is of no importance to the results. The heuristic results are compared to the model results in section 5.3.

### 5.3 Comparison and conclusion

In this section, selected results from the model and heuristic test phases are compared and some conclusive remarks on the different solution approaches are stated. The comparison of model and heuristic results includes medium-scale class instances solved by the CPPT model and the heuristic, not allowing rejections, respectively. Results from the CPP model tests are not considered here as the constraints do not include capacity restrictions.

Table 5.76 provides a basis for comparison of results for CPPT model and the heuristic when solving the medium-scale class problems. The scaled 8 and 10 containers instances are omitted as the CPPT model does not find a feasible solution within the time limit. In order to compare objective values, the original OPT and #M entries for the CPPT model are adjusted by #M and  $C$  respectively, as explained in section 5.1 on page 108, so that they are equivalent to the T and #M values for the heuristic, denoting the total transportation time and the accumulated number of full moves. The number of empty moves, previously stated as a part of the heuristic #M results, are omitted in table 5.76 as the model does not include empty moves. In the **Wait**, **Early**, and **Late** columns, the number of containers is left out, entries stating the total time for the respective time deviation. Note that when waiting time and/or delays occur in heuristic solutions, results cannot be compared directly to the model as this corresponds to a relaxation of the arrival and departure time constraints in the model. The final column, **Diff** states the difference in percent between the combined solution value,  $T + \#M$ , for the CPPT model and the heuristic, equal to  $((T^h + \#M^h) - (T^m + \#M^m)) / (T^m + \#M^m)$ , **m** and **h** denoting model and heuristic results respectively. Keys to the table are, apart from the above changes and additions, equivalent to previous result tables.

In the following sections some comparisons are made for each subclass. Important results commented in the text are highlighted in table 5.76. Finally, section 5.3.1 concludes the chapter.

| Medium<br>scale | The CPPT model |           |         |         | The heuristic |           |             |       |            |     | Diff<br>m/h   |
|-----------------|----------------|-----------|---------|---------|---------------|-----------|-------------|-------|------------|-----|---------------|
|                 | T              | #M        | GAP     | CPU*    | T             | #M        | Wait        | Early | Late       | CPU |               |
| 6S.t            | <b>22</b>      | <b>15</b> | 0.00    | 1,068.7 | <b>22.0</b>   | <b>15</b> | 0.0         | 0.0   | 1.0        | 0.0 | 0.00          |
| 6C.t            | <b>18</b>      | <b>12</b> | 0.00    | 2.2     | <b>18.0</b>   | <b>12</b> | 0.0         | 0.0   | 0.0        | 0.0 | 0.00          |
| 6M1.t           | 20             | 13        | 0.00    | 9.9     | 22.0          | 13        | 0.0         | 0.0   | 2.0        | 0.0 | 6.06          |
| 6M2.t           | <b>20</b>      | <b>12</b> | 0.00    | 9.4     | <b>20.0</b>   | <b>12</b> | 0.0         | 0.0   | 0.0        | 0.0 | 0.00          |
| 6M3.t           | 16             | 12        | 0.00    | 0.6     | 18.0          | 12        | 0.0         | 0.0   | 2.0        | 0.0 | 7.14          |
| 6M4.t           | <b>18</b>      | <b>12</b> | 0.00    | 1.2     | <b>18.0</b>   | <b>12</b> | 0.0         | 0.0   | 0.0        | 0.0 | 0.00          |
| 6M5.t           | <b>18</b>      | <b>12</b> | 0.00    | 1.9     | <b>18.0</b>   | <b>12</b> | 0.0         | 0.0   | 0.0        | 0.0 | 0.00          |
| 6M6.t           | <b>18</b>      | <b>12</b> | 0.00    | 7.7     | <b>18.0</b>   | <b>12</b> | 0.0         | 0.0   | 1.0        | 0.0 | 0.00          |
| 6S.s            | 30             | 21        | 103.54  | -       | 42.0          | 15        | <b>6.0</b>  | 0.0   | 0.0        | 0.1 | 11.76         |
| 6C.s            | 22             | 16        | 47.09   | -       | 32.0          | 12        | <b>6.0</b>  | 0.0   | 0.0        | 0.0 | 15.79         |
| 6M1.s           | 28             | 20        | 75.00   | -       | 42.0          | 13        | <b>20.0</b> | 0.0   | 0.0        | 0.0 | 14.58         |
| 6M2.s           | 24             | 16        | 41.67   | -       | 38.0          | 12        | <b>12.0</b> | 0.0   | 0.0        | 0.0 | 25.00         |
| 6M3.s           | 26             | 18        | 58.33   | -       | 34.0          | 12        | <b>6.0</b>  | 0.0   | 0.0        | 0.0 | 4.55          |
| 6M4.s           | 28             | 20        | 75.00   | -       | 34.0          | 12        | <b>8.0</b>  | 0.0   | 0.0        | 0.0 | <b>-4.17</b>  |
| 6M5.s           | <b>18</b>      | <b>12</b> | 1,235.4 | 1,235.6 | <b>34.0</b>   | <b>12</b> | <b>10.0</b> | 0.0   | 0.0        | 0.0 | 53.33         |
| 6M6.s           | <b>18</b>      | <b>12</b> | 1,743.9 | 1,744.1 | <b>32.0</b>   | <b>12</b> | <b>6.0</b>  | 0.0   | 0.0        | 0.0 | 46.67         |
| 8S.t            | 34             | 25        | 41.67   | -       | 32.0          | 20        | 0.0         | 0.0   | 15.0       | 0.0 | <b>-11.86</b> |
| 8C.t            | 28             | 18        | 5.47    | -       | 30.0          | 16        | 0.0         | 0.0   | 15.0       | 0.0 | <b>0.00</b>   |
| 8M1.t           | 30             | 18        | 11.11   | -       | 32.0          | 16        | 2.0         | 0.0   | 4.0        | 0.0 | <b>0.00</b>   |
| 8M2.t           | 28             | 19        | 8.12    | -       | 32.0          | 16        | 1.0         | 0.0   | 2.0        | 0.0 | <b>2.13</b>   |
| 8M3.t           | 30             | 20        | 16.46   | -       | 32.0          | 16        | 0.0         | 0.0   | 13.0       | 0.0 | <b>-4.00</b>  |
| 8M4.t           | 26             | 19        | 0.00    | 163.7   | 30.0          | 16        | 0.0         | 0.0   | 17.0       | 0.0 | 2.22          |
| 8M5.t           | 26             | 19        | 0.00    | 357.4   | 30.0          | 16        | 1.0         | 0.0   | 17.0       | 0.0 | 2.22          |
| 8M6.t           | 28             | 17        | 0.00    | 201.1   | 30.0          | 16        | 0.0         | 0.0   | 15.0       | 0.0 | 2.22          |
| 8M7.t           | 28             | 18        | 0.00    | 681.7   | 32.0          | 16        | 0.0         | 0.0   | 20.0       | 0.0 | 4.35          |
| 8M8.t           | <b>28</b>      | <b>18</b> | 0.00    | 1,432.3 | <b>28.0</b>   | <b>16</b> | 0.0         | 0.0   | <b>5.0</b> | 0.0 | <b>-4.35</b>  |
| 10S.t           | 40             | 28        | 43.08   | -       | 40.0          | 24        | 6.0         | 2.0   | 9.0        | 0.0 | <b>-5.88</b>  |
| 10C.t           | 50             | 32        | 130.76  | -       | 38.0          | 20        | 0.0         | 0.0   | 0.0        | 0.0 | <b>-29.27</b> |
| 10M1.t          | 40             | 27        | 42.36   | -       | 38.0          | 20        | 2.0         | 0.0   | 2.0        | 0.0 | <b>-13.43</b> |
| 10M2.t          | 40             | 23        | 32.47   | -       | 42.0          | 21        | 3.0         | 0.0   | 4.0        | 0.0 | 0.00          |
| 10M3.t          | 36             | 21        | 17.23   | -       | 38.0          | 20        | 0.0         | 0.0   | 0.0        | 0.0 | 1.75          |
| 10M4.t          | 46             | 29        | 66.16   | -       | 38.0          | 21        | 0.0         | 0.0   | 0.0        | 0.0 | <b>-21.33</b> |
| 10M5.t          | 36             | 25        | 26.82   | -       | 40.0          | 22        | 15.0        | 0.0   | 10.0       | 0.0 | 1.64          |
| 10M6.t          | 34             | 21        | 12.47   | -       | 40.0          | 21        | 4.0         | 0.0   | 0.0        | 0.0 | 10.91         |
| 10M7.t          | 32             | 21        | 7.08    | -       | 40.0          | 20        | 2.0         | 0.0   | 2.0        | 0.0 | 13.21         |
| 10M8.t          | 42             | 26        | 44.94   | -       | 38.0          | 21        | 0.0         | 0.0   | 1.0        | 0.0 | <b>-13.24</b> |
| 10M9.t          | 46             | 29        | 109.88  | -       | 40.0          | 20        | 0.0         | 0.0   | 1.0        | 0.0 | <b>-20.00</b> |
| 10M10.t         | 50             | 32        | 132.00  | -       | 38.0          | 20        | 0.0         | 0.0   | 0.0        | 0.0 | <b>-29.27</b> |

Table 5.76: Comparison of the results for selected medium-scale class instances solved by the CPPT model and the heuristic. Entries in the T and #M columns for the CPPT model are adjusted to be equivalent to those for the heuristic, representing the total transportation time and the total number of full moves. Table values in bold font are commented in the text.

## 6 containers subclass results

As all of the tight 6 containers instances are solved to optimality by the CPPT model, the heuristic results can be compared to optimum for these cases. In six of the eight instances, the heuristic reaches optimum in terms of total transportation time as well as the number of full moves and in the 6M1.t and 6M3.t instances, T values are 10 % and 12.5 % from optimum whereas #M values are equal to the CPPT results. Except for the stair instance, run times are comparable, not exceeding 10 seconds for the model.

Only two of the scaled 6 containers instances are solved to optimality by the CPPT model within the time limit, in both cases the total transportation time is about twice as long in the heuristic solution whereas the number of moves are identical for the two solution approaches. However, run times are close to the limit of 1,800 seconds for the model against less than 0.1 seconds for the heuristic. For the remaining scaled instances, GUB for the model results is compared to the heuristic outcome, as optimum is not reached within the time limit. In general, the model results in shorter transportation times whereas the heuristic produces a smaller number of moves. Adding the T and #M values provides a basis for comparison of performance, resulting in the summed heuristic solution values exceeding the model optimum or GUB by 4.5 - 53.3 %, except for the 6M4.s instance where the heuristic outperforms the model. Furthermore, some waiting time occurs in the heuristic results.

## 8 containers subclass results

In the 8 containers subclass, only half of the instances are solved to optimality by the CPPT model. In these cases, the heuristic results are very close to optimum, again in general, with larger T values and smaller #M values. When summing them, the model results outperform the heuristic by only a few percent. The 8M8.t instance is an exception to the other 8 containers problems which are solved to optimality by the CPPT model as the heuristic actually finds a better solution in terms of transportation time than the model does, the explanation being the three delayed containers, not allowed in the model solution. For the remaining tight instances in the 8 containers subclass, not solved to optimality by the CPPT model, the heuristic algorithm produces results equal or close to GUB in three cases and outperforms the model results in two runs, again, especially the number of full moves are low. However, some extent of waiting and delay occurs in several of the heuristic solutions.

## 10 containers subclass results

None of the 10 containers instances are solved to optimality by the CPPT model and heuristic run times are still below 0.1 seconds. In more than half of the solutions, the heuristic provides better results in terms of added T and #M values than the model. In these seven instances, the heuristic outperforms the model by 5.9 - 29.3 %. In one instance, the two approaches perform equally good, and in the remaining three instances, the model does up to 13.2 % better than the heuristic which is not impressive considering the difference in run times.

### 5.3.1 Conclusive remarks

In general, comparing the performance of the CPPT model to the heuristic algorithm leads to the conclusion that the heuristic approach to solving the problem considered is more suitable than the model approach.

As concluded in section 5.1.5 on page 128, the CPP model outperforms the CPPT model when increasing the problem size but the CPPT model, being more true to real-life restrictions by including capacity constraints, is the one being compared to the heuristic algorithm. However, not being able to solve several tight instances with just 6 - 10 containers within a 1,800 seconds time limit and running out of memory even before finding a feasible solution to scaled instances from the same subclasses, the CPPT model does not constitute a viable approach to the problem considered in this study.

Investigation of alternative ways to use the model(s) has not resulted in convincing arguments for applying different strategies for using a model approach. Extending the run time limit has a marginal effect on the solution quality, indicating that it is not worthwhile to obtain information long before solutions are required, and omitting part of the constraints implies far from feasible solutions and no significant gain in smaller gaps and run times in return, for which reason a “relaxation, separation, and adding violated constraints ad hoc” approach does not seem promising.

Thus, the models serve mainly as a tool for analyzing and appreciating the problem, its structure, and complexity. Further research concerning use of the models for exact solution of larger problem instances may concern construction of a problem-specific algorithm as alternative to using a standard solver. This possibly very fruitful direction is left for future research.

Contrary to model results, the heuristic algorithm produces good solutions within very short run times. Comparing to model solutions, the heuristic is able to find optimal - or close to optimal - objective values,  $T$  and  $\#M$ , indicating its suitability to produce solutions with short transportation times and, especially, few reshuffles. On the other hand, punctuality is not directly minimized by the heuristic procedures which makes the approach less appropriate, in its present form, if this is the primary goal.

It should, however, be noted that the standard of reference for comparing the model and heuristic approaches is quite limited as the CPPT model does not scale very well and smaller problem instances provide a weak basis for comparison.

Considering the heuristic results separately, there are several ways to control the algorithm so that it suits certain goals. As concluded in section 5.2.4 on page 202, if seeking to minimize the transportation times and the number of reshuffles, a small load factor, an oblong block, and a full range position search strategy should be applied. If, on the other hand, seeking to minimize deviations from the scheduled arrival and departure times, a larger load factor and a more square block provides better results. However, if not accepting waiting time and late departures, the algorithm must be adjusted to consider this as well. Furthermore, the proposed shifting routine, not being able to improve large-scale solutions, needs being further developed or reorganized if pursuing this direction.

The primary advantage of the heuristic lies in producing, in general, high-quality

solutions in very short computation times, also leaving considerable time to perform re-runs with alternative search strategies or new routines if desired. Concluding from the research performed in this study, it seems that solving a compact formulation of the CPP is not an efficient approach, leaving us with the question: Is it possible to find a better compact formulation of the problem, which enables finding optimal solution in short run times? It cannot be disproven based on this study but it seems unlikely due to the lack of structure, the high degree of freedom, and the symmetry of the problem. However, the proposed heuristic algorithm may in any case outperform exact approaches, especially when considering practical application where run times are of great importance. Further developments of the heuristic procedures and improvement routines may lead to even better solutions and implementation of different planning strategies may lead to promising results and a wide scope of application.





## Chapter 6

# Conclusions

The motivation of this PhD study was defined as the following. First, to investigate the potential of finding good mathematical optimization models for the container positioning problem (CPP), perform thorough analyses of the problem complexity, and explore exact solution approaches based on the models. Second, to develop an efficient heuristic algorithm by which large-scale problem instances can be solved in resonable run times.

Three new mathematical models have been proposed in this thesis. The first serving as a conceptual basis for the further work and the latter two models for the CPP representing two different formulations of the problem. Computational results confirm the indications from the model analyses: That increased problem sizes leads to weak lower bounds and large optimality gaps - if even finding feasible solutions - for which reason solving the models with standard optimizers seems unfit for addressing the CPP. Even though the use of standard optimization software presumably produces poorer results than if using problem-specific routines, it is believed that compact fomulations of the CPP will not provide an efficient approach to solving large-scale problem instances. On the other hand, mathematical models constitute an important instrument for understanding and appreciating the problem and, furthermore, form the basis for developing correct and well-performing solution algorithms.

Based on partly the understanding of the problem provided by the mathematical formulations, and partly the practical knowledge provided by correspondance with the industry, a heuristic algorithm and an improvement routine have been developed, proving to be a very efficient solution approach to the CPP. Not significantly affected by increase in problem size, the heuristic solution procedure provides high-quality results in very short run times. However, certain issues are not efficiently dealt with in the heuristic algorithm, punctuality being the most important one. If considering other objectives than the ones targeted by the models - minimizing reshuffling and transportation time - modifications must be implemented.

Concluding based on the computational results, the heuristic algorithm outperforms the models solved by standard software, especially if considering a practical application purpose. Even though a limited scope for comparison of the solution approaches blurs the picture to some extent, there are clear indications of the superiority of the heuristic

as incorporating additional features will not cause the algorithm to run much slower due to the design of the method.

However, both the research into the mathematical models and the work on the heuristic algorithm presented in this thesis suggest interesting perspectives for conducting future research in this area. Some key issues are proposed in the following section.

## 6.1 Perspectives

There are several perspectives for further research concerning the CPP. Three directions of future investigation are suggested: Compact formulations, exact solution algorithms, and further development of heuristic routines.

### Compact mathematical formulations

A good compact model for a problem may enable very efficient solution approaches by standard optimization software or relatively simple algorithms. This, however, depends on the structure and scalability of the model and it is not given that such a formulation can be achieved.

A fruitful aspect to investigate may be to construct an optimization model based on a formulation of well-known problems and additional constraints that may be relaxed, thus leaving a subproblem for which efficient solution techniques are available. Clearly, the challenge includes both the transformation to a well-known problem that captures essential parts of the CPP and the formulation of additional - possibly complicating - constraints in a way that makes their relaxation tractable.

There are some similarities between the CPP and the vehicle routing problem (VRP), but the lifo and capacity constraints constitute a significant difference between the two problems as these restrictions link together the container paths through the network. It may, however, turn out that future work will result in better models than the ones suggested in this thesis, possibly building on VRP formulations which may enable efficient solution of a compact formulation.

Alternatively, leaving out variables may lead to significant reductions in model size and, thereby, computation time. The CTP and CPP models contain a large number of lifo variables of which many are presumably redundant as few containers actually conflict in practice. The CPPT model consists of a large number of variables due to the time-discretized approach for which reason omitting part of them may be advantageous (see discussion in section 2.3.2 on page 56).

### Exact solution approaches

Whereas standard optimization software applying general routines that are valid for all problems of a given type often show to be ineffective for solving complex and/or large-scale problems, algorithms that are tailored for a specific problem may be far more efficient, resulting in much shorter computation times.

A first step towards constructing a problem-specific algorithm may be implementation of a compact formulation in a standard programming language, representing the problem matrix in a concise way and controlling the solution process effectively. Several platforms for linking standard code with a wide range of solvers exist.

Subsequently, upon investigating the solution performance of an efficient model implementation, a problem-specific algorithm to solve the problem even more effectively is an obvious avenue to explore. As discussed in the previous section, relaxing part of the variables and/or constraints may reduce the model size significantly, and if being able to solve the remaining subproblem efficiently there is clearly a potential in developing a separation routine and add required variables or violated constraints ad hoc or, alternatively, to apply a decomposition approach such as Lagrangian relaxation.

### **Heuristic algorithms**

Based on the work documented in this thesis, it is believed that a heuristic solution approach is the best suited for the CPP. The presented framework is based on a greedy algorithm and some well-chosen rules and criteria by which high-quality solutions are obtained in very short run times.

Some perspectives for improving the proposed algorithm include further development of the improvement routine which is not able to repair large-scale problem solutions in its present form, and investigation of a range of control strategies such as extending the concept of the event structure, adding different possibilities for varying positioning and reshuffling strategies, and treating punctuality explicitly rather than minimizing only transportation time and number of reshuffles.

Furthermore, as the CPP is a subproblem of the entire port container terminal problem, the presented heuristic algorithm constitutes a module in a complete terminal management system. Extending the concepts of the heuristic to a planning tool for the entire container terminal may be of great value to port authorities and terminal managers.

An alternative approach may concern combining the heuristic algorithm and an exact solution method, a technique called local branching, proposed by Fischetti and Lodi [17], the idea being to embed heuristic results in an exact solution method and accommodate the degree of change performed by the the exact algorithm, thereby drawing on both the heuristic procedure's strength in solving large-scale problems and the exact algorithm's ability to find good solutions.

Concluding, the perspectives for further development of heuristic approaches for the CPP seem promising and recommendable in preference to other research directions, especially if also considering an industrial application aspect.



# Bibliography

- [1] E. K. Bish. A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research*, 144:83–107, 2003.
- [2] E. K. Bish, T.-Y. Leong, C.-L. Li, J. W. C. Ng, and D. Simchi-Levi. Analysis of a new vehicle scheduling and location problem. *Naval Research Logistics*, 48:363–385, 2001.
- [3] S. Bonder. Army operations research - historical perspectives and lessons learned. *Operations Research*, 50:25–34, 2002.
- [4] E. K. Burke and G. Kendall, editors. *Search Methodologies: Introductory tutorials in optimization and decision support*. Springer Science + Business Media, Inc., 233 Spring Street, New York, NY 10013, USA, 2005.
- [5] B. Cao and G. Uebe. An algorithm for solving capacitated multicommodity p-median transportation problems. *Journal of the Operational Research Society*, 44: 259–269, 1993.
- [6] P. Chen, Z. Fu, A. Lim, and B. Rodrigues. Port yard storage optimization. *IEEE Transactions on Automation Science and Engineering*, 1:26–37, 2004.
- [7] T. Chen. Yard operations in the container terminal - a study in the "unproductive moves". *Maritime Policy and Management*, 26:27–38, 1999.
- [8] U. N. conference on trade and development. Review of maritime transport 2006. UNCTAD secretariat, United Nations, New York and Geneva, 2006.
- [9] J.-F. Cordeau, M. Gaudioso, G. Laporte, and L. Moccia. The service allocation problem at the gioia tauro maritime terminal. *European Journal of Operational Research*, 176:1167–1184, 2007.
- [10] R. W. Cottle. George B. Dantzig: Operations research icon. *Operations Research*, 53:892–898, 2005.
- [11] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

- [12] G. B. Dantzig. Linear programming. In J. K. Lenstra, A. H. G. R. Kan, and A. Schrijver, editors, *History of Mathematical Programming: A Collection of Personal Reminiscences*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1991.
- [13] Dash Optimization homepage. <http://www.dashoptimization.com>, 2008.
- [14] B. de Castilho and C. F. Daganzo. Handling strategies for import containers at marine terminals. *Transportation Research part B*, 27:151–166, 1993.
- [15] P. Dell’Olmo and G. Lulli. Planning activities in a network of logistic platforms with shared resources. *Annals of Operations Research*, 129:155–169, 2004.
- [16] Eclipse homepage. <http://www.eclipse.org>, 2008.
- [17] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2003.
- [18] GAMS homepage. <http://www.gams.com>, 2008.
- [19] S. I. Gass. Great moments in history, 2002. Chronological list of important OR contributions in [22].
- [20] R. E. Gomory. Early integer programming. In J. K. Lenstra, A. H. G. R. Kan, and A. Schrijver, editors, *History of Mathematical Programming: A Collection of Personal Reminiscences*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1991.
- [21] K. B. Haley. War and peace: The first 25 years of OR in Great Britain. *Operations Research*, 50:82–88, 2002.
- [22] P. R. Horner. History in the making. *OR/MS Today*, 29:30–39, 2002.
- [23] ILOG homepage. <http://www.ilog.com>, 2008.
- [24] International Maritime Organization. [www.imo.org](http://www.imo.org), 2007.
- [25] International Organization for Standardization. [www.iso.org](http://www.iso.org), 2007.
- [26] J. Kang, M.-S. Oh, E. Y. Ahn, K. R. Ryu, and K. H. Kim. Planning for intra-block remarshalling in a container terminal. In *Advances in applied artificial intelligence*, volume 4031, pages 1211–1220, 2006.
- [27] J. Kang, K. R. Ryu, and K. H. Kim. Determination of storage locations for incoming containers of uncertain weight. In *Advances in applied artificial intelligence*, volume 4031, pages 1159–1168, 2006.
- [28] K. H. Kim and J. W. Bae. Re-marshalling export containers in port container terminals. *Computers & Industrial Engineering*, 35:655–658, 1998.

- [29] K. H. Kim and G.-P. Hong. A heuristic rule for relocating blocks. *Computers & Operations Research*, 33:940–954, 2006.
- [30] K. H. Kim and D. Y. Kim. Group storage methods at container port terminals. *American Society of Mechanical Engineers, Material Handling Division, MHD; The Materials Handling Engineering Division 75th Anniversary*, 2:15–20, 1994.
- [31] K. H. Kim and H. B. Kim. Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, 59:415–423, 1999.
- [32] K. H. Kim and J.-S. Lee. Satisfying constraints for locating export containers in port container terminals. In *Computational science and its application / ICCSA 2006, PT3*, volume 3982, pages 564–573, 2006.
- [33] K. H. Kim and K. T. Park. A note on a dynamic space-allocation method for outbound containers. *European Journal of Operational Research*, 148:92–101, 2003.
- [34] K. H. Kim, Y. M. Park, and K.-R. Ryu. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124:89–101, 2000.
- [35] E. Kozan. Optimising container transfers at multimodal terminals. *Mathematical and Computer Modelling*, 31:235–243, 2000.
- [36] E. Kozan and B. Casey. Alternative algorithms for the optimization of a simulation model of a multimodal container terminal. *Journal of the Operational Research Society*, 58:1203–1213, 2007.
- [37] E. Kozan and P. Preston. Mathematical modelling of container transfers and storage locations at seaport terminals. *OR Spectrum*, 28:519–537, 2006.
- [38] E. Kozan and P. Preston. Genetic algorithms to schedule container transfers at multimodal terminals. *International Transactions in Operational Research*, 6:311–329, 1999.
- [39] G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 109–128. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.
- [40] H. Larnder. The origin of operational research. *Operations Research*, 32:465–475, 1984.
- [41] L. H. Lee, E. P. Chew, K. C. Tan, and Y. Han. An optimization model for storage yard management in transshipment hubs. *OR Spectrum*, 28:539–561, 2006.
- [42] Y. Lee and N.-Y. Hsu. An optimization model for the container pre-marshalling problem. *Computers & Operations Research*, 34:3295–3313, 2007.



- [43] J. D. C. Little. Philip M. Morse and the beginnings. *Operations Research*, 50: 146–148, 2002.
- [44] Maersk Line. [www.maerskline.com](http://www.maerskline.com), 2007.
- [45] P. J. M. Meersmans and R. Dekker. Operations research supports container handling. Technical Report 22, Econometric Institute, 2001.
- [46] P. M. Morse. *In at the Beginning: A Physicist's Life*. MIT Press, Cambridge, Mass, 1977.
- [47] P. M. Morse and G. E. Kimball. *Methods of Operations Research*. The Technology Press of MIT, Cambridge, MA, John Wiley, New York, 1951.
- [48] F. Nørgaard and K. Gitter. Private conversations, 2005–2008.
- [49] M. Paolucci, V. Recagno, and S. Sacone. Designing container yard management systems. In *Proceedings of the International Conference on Computer Aided Design, Manufacture and Operation in The Railway and Other Advanced Mass Transit Systems*, pages 351–360, 1998.
- [50] M. Pfetsch and R. Borndörfer. Private conversations, 2005.
- [51] P. Preston and E. Kozan. An approach to determine storage locations of containers at seaport terminals. *Computers & Operations Research*, 28:983–995, 2001.
- [52] R. Stahlbock and S. Voß. Operations research at container terminals: A literature update. *OR Spectrum*, 30:1–52, 2007.
- [53] D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26:3–49, 2004.
- [54] Sun Java homepage. <http://java.sun.com>, 2008.
- [55] M. Taleb-Ibrahimi, B. de Castilho, and C. F. Daganzo. Storage space vs handling work in container terminals. *Transportation Research Part B*, 27:13–32, 1993.
- [56] The White Pass Container Route. [www.explorenorth.com](http://www.explorenorth.com), 2001.
- [57] United Nations Atlas of the Oceans. [www.oceansatlas.org](http://www.oceansatlas.org), 2007.
- [58] K. M. van Hee and R. J. Wijbrands. Decision support system for container terminal planning. *European Journal of Operational Research*, 34:262–272, 1988.
- [59] I. F. A. Vis and R. de Koster. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147:1–16, 2003.
- [60] S. Voß, R. Stahlbock, and R. Eisenberg. Private conversations, 2004.
- [61] Wikipedia, The Free Encyclopedia. [www.wikipedia.org](http://www.wikipedia.org), 2008.

- [62] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, Inc., 1998.
- [63] C. Zhang, J. Liu, Y.-W. Wan, K. G. Murty, and R. J. Linn. Storage space allocation in container terminals. *Transportation Research part B*, 37:883–903, 2003.



# Mathematical models and heuristic solutions for container positioning problems in port terminals

## Appendix

Louise K Sibbesen

Technical University of Denmark  
Department of Management Engineering

February 2008



# Appendix A

## Overview of OR developments

This chapter provides a background knowledge of the methodological approach of this thesis: Operations Research (OR). Since it is both interesting and important to know the history of the methodological research field - especially for practitioners for whom this is may be unknown area - the main part of the chapter allows the reader to lose oneself in the history of impressive accomplishments in the OR arena. A brief overview of heuristics is provided in the final section of the chapter.

### A.1 Background on OR

*“Operations Research is a scientific method of providing executive departments with a quantitative basis for decisions regarding operations under their control.”*

Morse and Kimball [47]

#### Early contributions

OR and optimization, as we know it today, originated during the years around the Second World War but the thought of approaching problems by mathematical models or equation systems traces several centuries back. In fact, the concept of optimization, i.e. minimizing (or maximizing), was conceived in 1665 by Newton, inventing the method for finding a minimum solution of a function. In 1736 Euler considered the well-known Königsberg Bridge Problem, Lagrange worked out Lagrangian multipliers in 1788, the method of least squares was invented by Gauss and Legendre in 1795, and an important contribution to OR, war gaming, was studied by von Reisswitz in 1811. In 1826 Fourier and Gauss simultaneously found out how to solve inequalities and linear equations respectively and in 1902 Farkas found solutions to inequality systems. The first to introduce scientific management was Taylor in 1890, and Pareto optimality, Markov chains and theory of probabilities (by Erlang) were all contributions from the first decade of the 20th century. In 1936 Motzkin presented the transposition theorem and Weiszfeld and Vazsonyi introduced the facility location problem. In 1939 Karush found the optimality

condition for constrained problems and, in the same year, Kantorovich, being the first in history, published a monograph on linear programming and optimization but it was neglected by the USSR because of the subject's obvious conflicts with the communistic ideology and did not surface until after the breakthrough in the West almost ten years later [19].

The first initiatives, leading directly to the success of OR approaches during the Second World War, include the 1934 committee for Scientific Survey of Air Defence in the UK Air Force, chaired by Henry Tizard and facing the problem of defence against impending German air attacks. In 1935 Robert Watson-Watt was assigned to the task of developing a "death ray" and though the posed problem proved infeasible, he contributed to development of equipments to location of aircrafts by radio - the radar. While Germany tested equipment and developed air tactics by participating in the Spanish Civil War in 1936, British scientists worked at full speed to catch up with the opponent. As a result of an unsuccessful air-defence exercise, carried out in 1938, superintendent of the Bawdsey Research Station, established two years earlier, A. P. Rowe proposed an immediate change of focus from a technical to an operational research of the radar systems. This new applied approach was designated "operational research" - thereby representing the first conscious OR studies - and two teams of scientists from the radar research group, the first under leadership of the young physicist E. C. Williams and the second chaired by G. A. Roberts, embarked upon the mission which eventually enabled the British defence against the opposed force at the outbreak of war in 1939. However, the superior German forces necessitated yet more research efforts, for which reason a number of scientists, including Williams and Harold Larnder, evolved methods to prediction of operations outcome, based on losses and replacement rates, presented to air chief marshal Hugh Dowding in a graph form which convinced the high-level military decision-makers to change the course of attack which would have been fatal for the Allies. In fact, as response to a farewell note from Larnder to Sir Hugh Dowding, turning over his command in 1941, the recognition from the chief marshal was "Thanks. This war will be won by science thoughtfully applied to operational needs. H. D." [40].

### **During the Second World War**

All these contributions, early as well as contemporary, scientific as well as practical, were important sources of inspiration to the people who further developed the future OR discipline during the time of the Second World War, a period of great progress and fundamental importance to the OR field. Contemporary with the forming of UK and US military OR groups, on the academic scene, Hitchcock presented the transportation problem in 1941, Morse, Rinehart, Koopman, and Kimball introduced search theory in 1942, and two years later von Neumann and Morgenstern presented game theory and utility theory [19].

During the war, many scientists joined the military OR groups in Great Britain, the United States, and Canada and gave support in tactical and operational decision problems, collaboration between British, American, and Canadian OR sections dating back to 1942. The pioneer OR groups were established from 1939 onwards, the first in

Great Britain under leadership of professor Patrick M. S. Blackett and the first in the United States chaired by Philip M. Morse who acknowledged the success of OR in the UK war operations and spoke for duplicating the approach in US military services - a request appreciated by captain Wilder Baker who had witnessed the work by the British navy. Blackett was a highly respected physicist and could count the Nobel Prize in 1948 among orders and honours awarded. Morse was a fiery soul who even persuaded the navy to let himself and the scientists under his command observe field operations and thereby gain knowledge of actual results of their plans. Under his leadership, the Operations Research Group (originally called the Anti-Submarine Warfare Operations Research Group in merit and in size to about a hundred OR scientists in 1945, contributing to awarding him the Presidential Medal of Merit in 1946. The important work of the two OR pioneers - and personal friends - Blackett and Morse earned them the status of “father of OR” in Great Britain and the United States respectively, both for their critical wartime contributions to military services, for establishing OR as a scientific discipline, and for being prime movers behind the later creation of the first British/American OR societies [21, 43, 46].

One of the scientists, joining the US military OR studies, George B. Dantzig, a PhD student of Jerzy Neymann during the war who solved two unsolved mathematical problems that he thought was home work, took leave of absence from his doctoral studies to serve as chief of the Combat Analysis Branch, a contribution for which he was awarded the Exceptional Civilian Service Medal in 1944. The work of Dantzig should soon prove to be ground-breaking for which reason he has become an icon in the OR world [10].

### **A few important postwar years with major discoveries**

Immediately after the Second World War a few very productive years generated research, discoveries, and developments, of great importance for the subsequent advances in the OR field. Though the term “Operations/Operational Research” dates back to the pre-war period, the first to introduce OR as an academic discipline were Philip M. Morse and George E. Kimball in their 1951 publication *Methods of Operations Research* (first a classified report in 1946) [19, 22].

After completing his doctorate in 1946 Dantzig turned down an offer of a position at Berkely in favour of a job as mathematical advisor at Pentagon where he contributed to development of linear programming and invented the Simplex Method to solve the problem of deployment of training and supply activities. In 1949 at the Conference on Activity Analysis of Production and Allocation, Dantzig presented five papers (of a total of 25) dealing with linear programming (the name shortened from “Programming in a Linear Structure” after recommendation from Tjalling Koopmans, editor of the proceedings volume), and introduction of the Simplex algorithm. During his time at the mathematics department of the RAND Corporation, a “think tank” set up in 1948 and sponsored by the Air Force, Dantzig’s research included game theory, linear programming theory, the Simplex Method, large-scale linear programming, linear programming under uncertainty, network optimization problems, integer linear programming and discrete



optimization. Some of his most widely known publications document the Generalized Simplex Method (with Alex Orden and Phillip Wolfe), the primal-dual algorithm for linear programs (with Ford and Ray Fulkerson), the Dual Simplex Method, the Dantzig-Wolfe decomposition method (with Phillip Wolfe), maximal network flows and the max-flow min-cut theorem (with Ray Fulkerson), and ground-braking work on the traveling salesman problem (with Ray Fulkerson and Selmer Johnson). In 1963 Dantzig published his significant monograph *Linear Programming and Extensions* with the famous preface opening line “The final test of a theory is its capacity to solve the problems which originated it.” [11]. For his major contributions, he received a large number of honors and awards, including eight honorary doctorates [10]. Three very - if not the most - important contributions to the OR field is, for certain, linear programming, integer programming, and the Simplex Method.

### **Linear programming and the Simplex Method**

Linear programming - that is, building mathematical models with variables, restrictions, and goals, and defining sequences of decisions to take in order to achieve the goals in the best possible way - is based on three indispensable components: Mathematical models, solution algorithms, and computers and software. George B. Dantzig who invented linear programming as we know it today in 1947 credits some of the great mathematicians, contributing to the early developments and achievements: von Neumann, Kantorovich, Leontief, Koopmans, and Hitchcock. Inspired by the 1932 Interindustry Input-Output Model for the American economy by Wassily Leontief (a work for which he received the Nobel Prize in 1976), Dantzig formulated a linear programming model for the time-staged deployment, training and logistical supply problem, posed to him by Hitchcock and Wood in the Pentagon immediately after the Second World War. The time-staged dynamic linear programming model had no objective function to begin with, which was the case for all linear programming approaches prior to 1947 since it was computationally impossible to solve such models without electronic computers. Then, mathematical models consisted of linear inequalities but no objective function. When stating an explicit objective function as replacement for the ad-hoc ground rules that, till then, had controlled decisions based on systems of linear equations and inequalities (i.e. mathematical models up to 1947), Dantzig was the first to build a linear programming optimization model [12].

With the purpose of investigating techniques to solve such optimization models Dantzig visited Tjalling Koopmans, at the time a mathematical economist at the University of Chicago, who became very enthusiastic about the ground-braking contribution of an objective function to be optimized. However, there were no methods to solve such systems at the time, and Dantzig embarked on developing an algorithm that could do the job which is how the Simplex Method, introduced in the summer 1947 and further developed under consultancy from the highly respected game theory researcher Johnny von Neumann, came to exist and soon proved it's superiority [12].

The first official presentation of linear programming and the Simplex Method was given by Dantzig at the 1949 Conference on Activity Analysis of Production and Allo-

cation, organized by Tjalling Koopmans and attended by many great researchers. Linear programming and the Simplex Method formed the basis for development of many mathematical programming areas, including nonlinear programming (initiated by the Karush-Kuhn-Tucker conditions in 1951), network flow theory (accelerated by Flood, Ford, Fulkerson, and others in the early 1950's), large-scale methods (originated by the Dantzig-Wolfe decomposition method, reported in papers from 1959 and 1960), stochastic programming (begun by Dantzig's 1955 paper on linear programming under uncertainty), integer programming (brought forth in 1958 by Gomory, the father of cutting planes), and polynomial-time algorithms [12].

### **Integer programming**

The early research into integer programming was carried out by the mathematician Ralph Gomory and originated by the problem of fractional solutions to linear programming problems, attended to during the Second World War. Gomory did not become acquainted with OR until after completion of his PhD studies in 1954, after which he was assigned to the physics branch of the Office of Naval Research, the neighbour of the OR group where he was able to spend a great deal of time due to his competence, enabling him to do his job at less than full time. Taking a preparatory course by the OR pioneer Alan Goldman in 1957, he quickly became enthusiastic about the rapidly growing field [20].

Back at Princeton late in 1957 Gomory worked out how to approach integer programming problems by use of linear programming. Knowing that integer solutions were obtainable for systems of linear equations, Gomory took the first steps towards solving systems of linear inequalities in integers as well and, thereby, solving the problem of fractional solutions to linear programming problems. Realizing that a fractional solution to a linear programming maximization problem of  $7\frac{1}{4}$  naturally yields an integer solution of at most 7, Gomory conceived in only a few days the concept of valid inequalities for integer linear programming problems. Early in 1958 he made the first official presentation of the cutting plane algorithm together with Albert W. Tucker [20].

Gomory continued his work on integer programming for multi-terminal network flow problems, the traveling salesman problem, and the cutting stock problem - the latter work together with Paul Gilmore earned the Lanchester Prize in 1963 and practical case studies led to significant savings for mill companies, adopting the Dantzig-Wolfe decomposition based optimization tools in their production. Together with Ellis Johnson Gomory contributed with several papers on integer programming and corner polyhedra until he from 1970 to 1989 occupied director positions at IBM, jobs leaving no time to continue his substantial research into integer programming. He has received eight awards and seven honorary degrees [20].

### **OR societies and the Cold War period**

After the Second World War, several military OR groups representing different services were formed and OR organizations were founded with the purpose of sharing experiences and discussing how to use the methods, developed for war efforts, to civil and industrial

purposes. The first of such societies promoted by Tizard, Goodeve, Gordon, and Blackett in particular was the British Operational Research Club, at the establishment in 1948 constituted of 30 scientists from the various military OR groups. Two years later the first issue of the Operational Research Quarterly was published by the OR Club [21].

In 1952 the Operations Research Society in America (ORSA) was founded, then consisting of 73 people from academia, industry, and the military, the first president being MIT professor Phillip M. Morse. Later that year the first issue of the ORSA journal Operations Research published. A group of ORSA members took the initiative to changing the focus from almost merely military matters to application of OR techniques to management problems, typically occurring in the industry. This resulted in the foundation of The Institute of Management Sciences (TIMS) in 1953 with William C. Cooper as the first president and the philosopher C. West Churchman as the first editor of the TIMS journal Management Science. ORSA and TIMS collaborated in the 1970's on the Interfaces publication, the first joint US meeting in 1974, followed by the establishment of the joint publication OR/MS Today, and in 1988 Saul Gass chaired the TIMS/ORSA National Meeting with a record of 2,879 participants but it was not until 1995 that ORSA and TIMS merged into the Institute for Operations Research and Management Sciences (INFORMS) and got the first common president John D. C. Little, former president of both ORSA and TIMS [22].

In September 1957 Oxford hosted an international conference on OR, sponsored by both the British and the US OR societies and TIMS, which subsequently led to establishment of OR societies in many countries. Due to the great success of the international conference, the International Federation of Operations Research Societies (IFORS) was established in January 1959, and one year later the second of the now triennial OR conference was held in France [21].

During the Cold War from 1950 through 1989 OR played an important role for the NATO alliance against the Warsaw Pact, greatly superior in conventional force capability but reluctant to attack due to the fragile nuclear weapon balance between the warring parties. The tension and the constant risk of war breaking out led to significant expansions in activities performed by military OR sections [3].

Some important achievements from the time of the Cold War include the Nash equilibrium, dynamic programming by Bellman, the first computer-solved transportation problem, nonlinear programming by Kuhn and Tucker, quadratic programming by Frank and Wolfe, and Dijkstra's shortest-route problem, all conducted in the 1950's, the branch and bound technique developed by Land, Doig, and others in 1960 Kwan's Chinese postman's problem from 1962, Clarke and Wright's 1964 vehicle routing savings algorithm, complexity theory by Edmonds and Karp one year later, the 1967 publications Introduction to Operations Research by Hillier and Lieberman and Theory of Scheduling by Conway, Maxwell and Miller, studies on computation complexity by Cook and Karp in 1971, Lagrangian relaxation by Geoffrion in 1978, constraint programming and parallel computing, revenue management in the airline industry by Cook, the simulated annealing and tabu search heuristics by Metropolis and Glover respectively, Karmarkar's interior point method, and Meeraus' General Algebraic Modeling System (GAMS) all

conceived in the 1980's. In addition, supply chain management introduced in 1990 should be mentioned [19].

## A.2 Heuristics and metaheuristics

This section provides a brief overview of heuristics and metaheuristics as solution approaches, alternative to pursuing a proven optimal solution. Heuristics are methods that solve a given problem to a certain satisfaction - but not necessarily to optimality - within relatively short time by applying a set of rules and educated guesses. Heuristic algorithms target a specific problem whereas metaheuristics apply heuristic principles for solving general classes of problems [61].

Some common metaheuristics and heuristic routines include local search, moving from solution to solution by evaluating a specified neighbourhood of candidates, greedy algorithms, iteratively making locally optimal choices in the search for global optimum, simulated annealing, applying “cooling” and “heating” techniques to intensify and diversify the search for global optimum by evaluating nearby solutions and regularly escaping local optima by allowing poor solutions, tabu search, searching neighbourhood solutions and seeking to avoid repeatedly returning to locally optimal solutions by marking a number of recently chosen ones “tabu”, and genetic algorithms, inspired by evolutionary biology, iteratively modifying current solutions to form new and better ones.

An overview of metaheuristics and heuristic techniques from 2005 is provided by Burke and Kendall [4].



# Appendix B

## Input data

All input data files can be found on the enclosed CD-rom.



## Appendix C

### CTP model solution

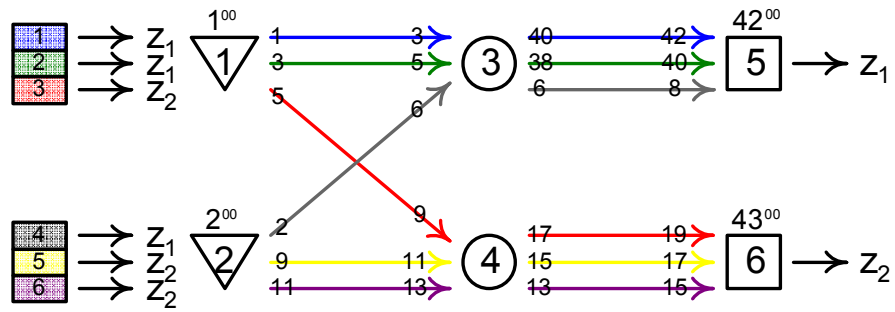


Figure C.1: Overview of the solution to the small validation test case with two ships, carrying three containers each, two storage positions, and two vehicles, destined for zone 1 and 2 respectively.



| S O L V E                |           | S U M M A R Y            |                                |        |        |        |
|--------------------------|-----------|--------------------------|--------------------------------|--------|--------|--------|
| MODEL                    | csp       | OBJECTIVE                | Total                          |        |        |        |
| TYPE                     | MIP       | DIRECTION                | MINIMIZE                       |        |        |        |
| SOLVER                   | CPLEX     | FROM LINE                | 200                            |        |        |        |
| **** SOLVER STATUS       |           | 1                        | NORMAL COMPLETION              |        |        |        |
| **** MODEL STATUS        |           | 1                        | OPTIMAL                        |        |        |        |
| **** OBJECTIVE VALUE     |           | 28.0000                  |                                |        |        |        |
| RESOURCE USAGE, LIMIT    |           | 0.130                    | 1000.000                       |        |        |        |
| ITERATION COUNT, LIMIT   |           | 66                       | 10000                          |        |        |        |
| MODEL STATISTICS         |           |                          |                                |        |        |        |
| BLOCKS OF EQUATIONS      | 19        | SINGLE EQUATIONS         | 855                            |        |        |        |
| BLOCKS OF VARIABLES      | 6         | SINGLE VARIABLES         | 433                            |        |        |        |
| NON ZERO ELEMENTS        | 7753      | DISCRETE VARIABLES       | 396                            |        |        |        |
| GENERATION TIME          | =         | 0.088 SECONDS            | 4.0 Mb SOL215-140 Nov 11, 2004 |        |        |        |
| EXECUTION TIME           | =         | 0.089 SECONDS            | 4.0 Mb SOL215-140 Nov 11, 2004 |        |        |        |
| Proven optimal solution. |           |                          |                                |        |        |        |
| MIP Solution:            | 28.000000 | (66 iterations, 0 nodes) |                                |        |        |        |
| Final Solve:             | 28.000000 | (0 iterations)           |                                |        |        |        |
| Best possible:           | 28.000000 |                          |                                |        |        |        |
| Absolute gap:            | 0.000000  |                          |                                |        |        |        |
| Relative gap:            | 0.000000  |                          |                                |        |        |        |
| X(c,i,j)                 | 3         | 4                        | 5                              | 6      |        |        |
| 1.1                      | 1.000     |                          |                                |        |        |        |
| 1.3                      |           |                          | 1.000                          |        |        |        |
| 2.1                      | 1.000     |                          |                                |        |        |        |
| 2.3                      |           |                          | 1.000                          |        |        |        |
| 3.1                      |           | 1.000                    |                                |        |        |        |
| 3.4                      |           |                          |                                | 1.000  |        |        |
| 4.2                      | 1.000     |                          |                                |        |        |        |
| 4.3                      |           |                          | 1.000                          |        |        |        |
| 5.2                      |           | 1.000                    |                                |        |        |        |
| 5.4                      |           |                          |                                | 1.000  |        |        |
| 6.2                      |           | 1.000                    |                                |        |        |        |
| 6.4                      |           |                          |                                | 1.000  |        |        |
| T(c,i)                   | 1         | 2                        | 3                              | 4      | 5      | 6      |
| 1                        | 1.000     |                          | 40.000                         |        | 42.000 |        |
| 2                        | 3.000     |                          | 38.000                         |        | 42.000 |        |
| 3                        | 5.000     |                          |                                | 17.000 |        | 43.000 |
| 4                        |           | 2.000                    | 6.000                          |        | 42.000 |        |
| 5                        |           | 9.000                    |                                | 15.000 |        | 43.000 |
| 6                        |           | 11.000                   |                                | 13.000 |        | 43.000 |

Table C.1: Solution to the small test case run by the CTP model.

# Appendix D

## CPP model solutions

### D.1 Small-scale class instances

|                                                       |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|-------------------------------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| OBJECTIVE: 13                                         |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Total number container positionings (sum of xp's) = 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Total moving time (sum of xm's) = 10                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Sequence                                              |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c   1 2 3                                             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -----                                                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1   0 2 3                                             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2   0 1 3                                             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3   0 2 3                                             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Positioning, storage, and moving times                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| n   Number: 1   Number: 2   Number: 3                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| c   i : p s m   i : p s m   i : p s m                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -----                                                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1   0 : 1 0 2   2 : 3 2 2   3 : 7 0 0                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2   0 : 2 0 1   1 : 3 5 1   3 : 9 0 0                 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3   0 : 3 0 2   2 : 5 4 2   3 : 11 0 0                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Table D.1: CPP model solution of the 3S.t instance.

```

OBJECTIVE: 9
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 6

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 7 1 | 3 : 10 0 0 |
2 | 0 : 2 0 1 | 1 : 3 5 1 | 3 : 9 0 0 |
3 | 0 : 3 0 1 | 1 : 4 3 1 | 3 : 8 0 0 |

```

Table D.2: CPP model solution of the 3C.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 2 2 | 3 : 7 0 0 |
2 | 0 : 2 0 1 | 1 : 3 8 1 | 3 : 12 0 0 |
3 | 0 : 3 0 1 | 1 : 4 5 1 | 3 : 10 0 0 |

```

Table D.3: CPP model solution of the 3M1A.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 4 2 | 3 : 9 0 0 |
2 | 0 : 3 0 1 | 1 : 4 7 1 | 3 : 12 0 0 |
3 | 0 : 5 0 1 | 1 : 6 4 1 | 3 : 11 0 0 |

```

Table D.4: CPP model solution of the 3M1B.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 1 3
3 | 0 2 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 4 1 | 3 : 7 0 0 |
2 | 0 : 2 0 1 | 1 : 3 2 1 | 3 : 6 0 0 |
3 | 0 : 3 0 2 | 2 : 5 2 2 | 3 : 9 0 0 |

```

Table D.5: CPP model solution of the **3M2A.t** instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 2 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 6 1 | 3 : 9 0 0 |
2 | 0 : 2 0 2 | 2 : 4 0 2 | 3 : 6 0 0 |
3 | 0 : 3 0 1 | 1 : 4 3 1 | 3 : 8 0 0 |

```

Table D.6: CPP model solution of the **3M2B.t** instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 5 2 | 3 : 10 0 0 |
2 | 0 : 2 0 1 | 1 : 3 8 1 | 3 : 12 0 0 |
3 | 0 : 3 0 1 | 1 : 4 3 1 | 3 : 8 0 0 |

```

Table D.7: CPP model solution of the **3M3A.t** instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 2 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 7 1 | 3 : 10 0 0 |
2 | 0 : 3 0 2 | 2 : 5 4 2 | 3 : 11 0 0 |
3 | 0 : 5 0 1 | 1 : 6 1 1 | 3 : 8 0 0 |

```

Table D.8: CPP model solution of the 3M3B.t instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xp's) = 4
Total moving time (sum of xm's) = 10

Sequence
c | 1 2 3 4
-----
1 | 0 2 1 3
2 | 0 1 3
3 | 0 2 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 0 1 | 1 : 4 16 1 | 3 : 21 0 0 |
2 | 0 : 2 0 1 | 1 : 3 23 1 | 3 : 27 0 0 |
3 | 0 : 3 0 2 | 2 : 5 26 2 | 3 : 33 0 0 |

```

Table D.9: CPP model solution of the 3S.s instance.

```

OBJECTIVE: 9
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 6

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 27 1 | 3 : 30 0 0 |
2 | 0 : 2 0 1 | 1 : 3 23 1 | 3 : 27 0 0 |
3 | 0 : 3 0 1 | 1 : 4 19 1 | 3 : 24 0 0 |

```

Table D.10: CPP model solution of the 3C.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 16 2 | 3 : 21 0 0 |
2 | 0 : 2 0 1 | 1 : 3 32 1 | 3 : 36 0 0 |
3 | 0 : 3 0 1 | 1 : 4 25 1 | 3 : 30 0 0 |

```

Table D.11: CPP model solution of the 3M1A.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 22 2 | 3 : 27 0 0 |
2 | 0 : 3 0 1 | 1 : 4 31 1 | 3 : 36 0 0 |
3 | 0 : 5 0 1 | 1 : 6 26 1 | 3 : 33 0 0 |

```

Table D.12: CPP model solution of the 3M1B.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 1 3
3 | 0 2 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 18 1 | 3 : 21 0 0 |
2 | 0 : 2 0 1 | 1 : 3 14 1 | 3 : 18 0 0 |
3 | 0 : 3 0 2 | 2 : 5 20 2 | 3 : 27 0 0 |

```

Table D.13: CPP model solution of the 3M2A.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 2 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 24 1 | 3 : 27 0 0 |
2 | 0 : 2 0 2 | 2 : 4 12 2 | 3 : 18 0 0 |
3 | 0 : 3 0 1 | 1 : 4 19 1 | 3 : 24 0 0 |

```

Table D.14: CPP model solution of the 3M2B.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 25 2 | 3 : 30 0 0 |
2 | 0 : 2 0 1 | 1 : 3 32 1 | 3 : 36 0 0 |
3 | 0 : 3 0 1 | 1 : 4 19 1 | 3 : 24 0 0 |

```

Table D.15: CPP model solution of the 3M3A.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xp's) = 3
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 25 2 | 3 : 30 0 0 |
2 | 0 : 3 0 1 | 1 : 4 28 1 | 3 : 33 0 0 |
3 | 0 : 5 0 1 | 1 : 6 17 1 | 3 : 24 0 0 |

```

Table D.16: CPP model solution of the 3M3B.s instance.

```

OBJECTIVE: 20
Total number container positionings (sum of xp's) = 6
Total moving time (sum of xm's) = 14

Sequence
c | 1 2 3 4
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 2 3
4 | 0 1 2 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 7 2 | 3 : 12 0 0 |
2 | 0 : 3 0 1 | 1 : 4 9 1 | 3 : 14 0 0 |
3 | 0 : 5 0 1 | 1 : 6 4 1 | 2 : 11 3 2 | 3 : 16 0 0 |
4 | 0 : 7 0 1 | 1 : 8 1 1 | 2 : 10 6 2 | 3 : 18 0 0 |

```

Table D.17: CPP model solution of the 4S.t instance.

```

OBJECTIVE: 12
Total number container positionings (sum of xp's) = 4
Total moving time (sum of xm's) = 8

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 1 3
3 | 0 1 3
4 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 15 1 | 3 : 18 0 0 |
2 | 0 : 3 0 1 | 1 : 4 11 1 | 3 : 16 0 0 |
3 | 0 : 5 0 1 | 1 : 6 7 1 | 3 : 14 0 0 |
4 | 0 : 7 0 1 | 1 : 8 3 1 | 3 : 12 0 0 |

```

Table D.18: CPP model solution of the 4C.t instance.



OBJECTIVE: 16  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 12

Sequence  
c | 1 2 3  
-----  
1 | 0 2 3  
2 | 0 1 3  
3 | 0 1 3  
4 | 0 2 3

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 2 : 3 6 2 | 3 : 11 0 0 |  
2 | 0 : 3 0 1 | 1 : 4 10 1 | 3 : 15 0 0 |  
3 | 0 : 5 0 1 | 1 : 6 6 1 | 3 : 13 0 0 |  
4 | 0 : 7 0 2 | 2 : 9 6 2 | 3 : 17 0 0 |

Table D.19: CPP model solution of the 4M1A.t instance.

OBJECTIVE: 17  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 12

Sequence  
c | 1 2 3 4  
-----  
1 | 0 1 2 3  
2 | 0 2 3  
3 | 0 1 3  
4 | 0 1 3

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
c | i : p s m | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 1 : 2 2 1 | 2 : 5 3 2 | 3 : 10 0 0 |  
2 | 0 : 3 0 2 | 2 : 5 7 2 | 3 : 14 0 0 |  
3 | 0 : 5 0 1 | 1 : 6 9 1 | 3 : 16 0 0 |  
4 | 0 : 7 0 1 | 1 : 8 3 1 | 3 : 12 0 0 |

Table D.20: CPP model solution of the 4M1B.t instance.

OBJECTIVE: 17  
 Total number container positionings (sum of xp's) = 5  
 Total moving time (sum of xm's) = 12

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 3 |   |
| 2 | 0 | 1 | 3 |   |
| 3 | 0 | 2 | 3 |   |
| 4 | 0 | 2 | 1 | 3 |

Positioning, storage, and moving times  

| n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |   |   |   |   |   |   |    |   |   |    |   |   |
|---|-----------|-----------|-----------|-----------|---|---|---|---|---|---|---|---|---|---|----|---|---|----|---|---|
| c | i         | p         | s         | m         | i | p | s | m | i | p | s | m | i | p | s  | m |   |    |   |   |
| 1 | 0         | :         | 1         | 0         | 1 | : | 1 | : | 2 | 9 | 1 | : | 3 | : | 12 | 0 | 0 |    |   |   |
| 2 | 0         | :         | 3         | 0         | 1 | : | 1 | : | 4 | 5 | 1 | : | 3 | : | 10 | 0 | 0 |    |   |   |
| 3 | 0         | :         | 5         | 0         | 2 | : | 2 | : | 7 | 5 | 2 | : | 3 | : | 14 | 0 | 0 |    |   |   |
| 4 | 0         | :         | 7         | 0         | 2 | : | 2 | : | 9 | 2 | 1 | : | 1 | : | 12 | 3 | 1 |    |   |   |
|   |           |           |           |           |   |   |   |   |   |   |   |   |   |   |    | 3 | : | 16 | 0 | 0 |

Table D.21: CPP model solution of the 4M2A.t instance.

OBJECTIVE: 16  
 Total number container positionings (sum of xp's) = 4  
 Total moving time (sum of xm's) = 12

Sequence  

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 2 | 0 | 2 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 1 | 3 |

Positioning, storage, and moving times  

| n | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |   |   |   |   |   |   |   |    |   |   |
|---|-----------|-----------|-----------|---|---|---|---|---|---|---|---|---|---|---|----|---|---|
| c | i         | p         | s         | m | i | p | s | m | i | p | s | m |   |   |    |   |   |
| 1 | 0         | :         | 1         | 0 | 2 | : | 2 | : | 3 | 9 | 2 | : | 3 | : | 14 | 0 | 0 |
| 2 | 0         | :         | 3         | 0 | 2 | : | 2 | : | 5 | 3 | 2 | : | 3 | : | 10 | 0 | 0 |
| 3 | 0         | :         | 5         | 0 | 1 | : | 1 | : | 6 | 9 | 1 | : | 3 | : | 16 | 0 | 0 |
| 4 | 0         | :         | 7         | 0 | 1 | : | 1 | : | 8 | 3 | 1 | : | 3 | : | 12 | 0 | 0 |

Table D.22: CPP model solution of the 4M2B.t instance.

OBJECTIVE: 16  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 12

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 2 | 3 |
| 4 | 0 | 1 | 3 |

| Positioning, storage, and moving times |           |   |   |   |           |   |   |   |           |    |   |   |   |   |    |   |   |
|----------------------------------------|-----------|---|---|---|-----------|---|---|---|-----------|----|---|---|---|---|----|---|---|
| n                                      | Number: 1 |   |   |   | Number: 2 |   |   |   | Number: 3 |    |   |   |   |   |    |   |   |
| c                                      | i         | p | s | m | i         | p | s | m | i         | p  | s | m |   |   |    |   |   |
| 1                                      | 0         | : | 1 | 0 | 2         | : | 2 | : | 3         | 7  | 2 | : | 3 | : | 12 | 0 | 0 |
| 2                                      | 0         | : | 3 | 0 | 1         | : | 1 | : | 4         | 11 | 1 | : | 3 | : | 16 | 0 | 0 |
| 3                                      | 0         | : | 5 | 0 | 2         | : | 2 | : | 7         | 1  | 2 | : | 3 | : | 10 | 0 | 0 |
| 4                                      | 0         | : | 7 | 0 | 1         | : | 1 | : | 8         | 5  | 1 | : | 3 | : | 14 | 0 | 0 |

Table D.23: CPP model solution of the 4M3A.t instance.

OBJECTIVE: 14  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 10

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 2 | 3 |

| Positioning, storage, and moving times |           |   |   |   |   |           |   |   |   |    |           |   |   |   |    |   |   |
|----------------------------------------|-----------|---|---|---|---|-----------|---|---|---|----|-----------|---|---|---|----|---|---|
| n                                      | Number: 1 |   |   |   |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |    |   |   |
| c                                      | i         | p | s | m |   | i         | p | s | m |    | i         | p | s | m |    |   |   |
| 1                                      | 0         | : | 1 | 0 | 1 | :         | 1 | : | 2 | 12 | 1         | : | 3 | : | 15 | 0 | 0 |
| 2                                      | 0         | : | 3 | 0 | 1 | :         | 1 | : | 4 | 6  | 1         | : | 3 | : | 11 | 0 | 0 |
| 3                                      | 0         | : | 5 | 0 | 1 | :         | 1 | : | 6 | 2  | 1         | : | 3 | : | 9  | 0 | 0 |
| 4                                      | 0         | : | 7 | 0 | 2 | :         | 2 | : | 9 | 2  | 2         | : | 3 | : | 13 | 0 | 0 |

Table D.24: CPP model solution of the 4M3B.t instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xp's) = 4
Total moving time (sum of xm's) = 10

Sequence
c | 1 2 3
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3
4 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 7 2 | 3 : 12 0 0 |
2 | 0 : 3 0 1 | 1 : 4 11 1 | 3 : 16 0 0 |
3 | 0 : 5 0 1 | 1 : 6 7 1 | 3 : 14 0 0 |
4 | 0 : 7 0 1 | 1 : 8 1 1 | 3 : 10 0 0 |

```

Table D.25: CPP model solution of the 4M4A.t instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xp's) = 4
Total moving time (sum of xm's) = 10

Sequence
c | 1 2 3
-----
1 | 0 1 3
2 | 0 2 3
3 | 0 1 3
4 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 15 1 | 3 : 18 0 0 |
2 | 0 : 3 0 2 | 2 : 5 7 2 | 3 : 14 0 0 |
3 | 0 : 5 0 1 | 1 : 6 9 1 | 3 : 16 0 0 |
4 | 0 : 7 0 1 | 1 : 8 3 1 | 3 : 12 0 0 |

```

Table D.26: CPP model solution of the 4M4B.t instance.

OBJECTIVE: 20  
 Total number container positionings (sum of xp's) = 6  
 Total moving time (sum of xm's) = 14

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 2 | 1 | 3 |
| 2 | 0 | 2 | 1 | 3 |
| 3 | 0 | 1 | 3 |   |
| 4 | 0 | 2 | 3 |   |

Positioning, storage, and moving times  

| n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |   |    |   |   |   |   |    |    |   |   |   |   |    |   |   |   |
|---|-----------|-----------|-----------|-----------|---|---|---|---|---|----|---|---|---|---|----|----|---|---|---|---|----|---|---|---|
| c | i         | p         | s         | m         | i | p | s | m | i | p  | s | m | i | p | s  | m  |   |   |   |   |    |   |   |   |
| 1 | 0         | :         | 1         | 0         | 2 | : | 2 | : | 3 | 3  | 1 | : | 1 | : | 7  | 28 | 1 | : | 3 | : | 36 | 0 | 0 | : |
| 2 | 0         | :         | 3         | 0         | 2 | : | 2 | : | 5 | 0  | 1 | : | 1 | : | 6  | 35 | 1 | : | 3 | : | 42 | 0 | 0 | : |
| 3 | 0         | :         | 5         | 0         | 1 | : | 1 | : | 6 | 41 | 1 | : | 3 | : | 48 | 0  | 0 | : |   |   |    |   |   |   |
| 4 | 0         | :         | 7         | 0         | 2 | : | 2 | : | 9 | 43 | 2 | : | 3 | : | 54 | 0  | 0 | : |   |   |    |   |   |   |

Table D.27: CPP model solution of the 4S.s instance.

OBJECTIVE: 12  
 Total number container positionings (sum of xp's) = 4  
 Total moving time (sum of xm's) = 8

Sequence  

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 1 | 3 |

Positioning, storage, and moving times  

| n | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |   |   |    |   |   |   |   |    |   |   |   |
|---|-----------|-----------|-----------|---|---|---|---|---|---|----|---|---|---|---|----|---|---|---|
| c | i         | p         | s         | m | i | p | s | m | i | p  | s | m |   |   |    |   |   |   |
| 1 | 0         | :         | 1         | 0 | 1 | : | 1 | : | 2 | 51 | 1 | : | 3 | : | 54 | 0 | 0 | : |
| 2 | 0         | :         | 3         | 0 | 1 | : | 1 | : | 4 | 43 | 1 | : | 3 | : | 48 | 0 | 0 | : |
| 3 | 0         | :         | 5         | 0 | 1 | : | 1 | : | 6 | 35 | 1 | : | 3 | : | 42 | 0 | 0 | : |
| 4 | 0         | :         | 7         | 0 | 1 | : | 1 | : | 8 | 27 | 1 | : | 3 | : | 36 | 0 | 0 | : |

Table D.28: CPP model solution of the 4C.s instance.

```

OBJECTIVE: 17
Total number container positionings (sum of xp's) = 5
Total moving time (sum of xm's) = 12

Sequence
c | 1 2 3 4
-----
1 | 0 2 3
2 | 0 1 3
3 | 0 1 3
4 | 0 1 2 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 28 2 | 3 : 33 0 0 |
2 | 0 : 3 0 1 | 1 : 4 40 1 | 3 : 45 0 0 |
3 | 0 : 5 0 1 | 1 : 6 32 1 | 3 : 39 0 0 |
4 | 0 : 7 0 1 | 1 : 8 22 1 | 2 : 31 18 2 | 3 : 51 0 0 |

```

Table D.29: CPP model solution of the 4M1A.s instance.

```

OBJECTIVE: 17
Total number container positionings (sum of xp's) = 5
Total moving time (sum of xm's) = 12

Sequence
c | 1 2 3 4
-----
1 | 0 2 3
2 | 0 2 1 3
3 | 0 1 3
4 | 0 1 3

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 2 : 3 25 2 | 3 : 30 0 0 |
2 | 0 : 3 0 2 | 2 : 5 2 1 | 1 : 8 33 1 | 3 : 42 0 0 |
3 | 0 : 5 0 1 | 1 : 6 41 1 | 3 : 48 0 0 |
4 | 0 : 7 0 1 | 1 : 8 27 1 | 3 : 36 0 0 |

```

Table D.30: CPP model solution of the 4M1B.s instance.

OBJECTIVE: 17  
 Total number container positionings (sum of xp's) = 5  
 Total moving time (sum of xm's) = 12

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 3 |   |
| 2 | 0 | 1 | 3 |   |
| 3 | 0 | 1 | 2 | 3 |
| 4 | 0 | 2 | 3 |   |

Positioning, storage, and moving times  

| n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |   |    |   |   |   |   |    |    |   |
|---|-----------|-----------|-----------|-----------|---|---|---|---|---|----|---|---|---|---|----|----|---|
| c | i         | p         | s         | m         | i | p | s | m | i | p  | s | m | i | p | s  | m  |   |
| 1 | 0         | :         | 1         | 0         | 1 | : | 1 | : | 2 | 33 | 1 | : | 3 | : | 36 | 0  | 0 |
| 2 | 0         | :         | 3         | 0         | 1 | : | 1 | : | 4 | 25 | 1 | : | 3 | : | 30 | 0  | 0 |
| 3 | 0         | :         | 5         | 0         | 1 | : | 1 | : | 6 | 2  | 1 | : | 2 | : | 9  | 31 | 2 |
| 4 | 0         | :         | 7         | 0         | 2 | : | 2 | : | 9 | 37 | 2 | : | 3 | : | 48 | 0  | 0 |

Table D.31: CPP model solution of the 4M2A.s instance.

OBJECTIVE: 16  
 Total number container positionings (sum of xp's) = 4  
 Total moving time (sum of xm's) = 12

Sequence  

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 2 | 0 | 2 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 1 | 3 |

Positioning, storage, and moving times  

| n | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |   |   |    |   |   |   |   |    |   |   |
|---|-----------|-----------|-----------|---|---|---|---|---|---|----|---|---|---|---|----|---|---|
| c | i         | p         | s         | m | i | p | s | m | i | p  | s | m |   |   |    |   |   |
| 1 | 0         | :         | 1         | 0 | 2 | : | 2 | : | 3 | 37 | 2 | : | 3 | : | 42 | 0 | 0 |
| 2 | 0         | :         | 3         | 0 | 2 | : | 2 | : | 5 | 23 | 2 | : | 3 | : | 30 | 0 | 0 |
| 3 | 0         | :         | 5         | 0 | 1 | : | 1 | : | 6 | 41 | 1 | : | 3 | : | 48 | 0 | 0 |
| 4 | 0         | :         | 7         | 0 | 1 | : | 1 | : | 8 | 27 | 1 | : | 3 | : | 36 | 0 | 0 |

Table D.32: CPP model solution of the 4M2B.s instance.

OBJECTIVE: 16  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 12

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 2 | 3 |
| 4 | 0 | 1 | 3 |

| Positioning, storage, and moving times |           |   |   |   |   |           |   |   |    |   |           |   |    |   |   |
|----------------------------------------|-----------|---|---|---|---|-----------|---|---|----|---|-----------|---|----|---|---|
| n                                      | Number: 1 |   |   |   |   | Number: 2 |   |   |    |   | Number: 3 |   |    |   |   |
| c                                      | i         | : | p | s | m | i         | : | p | s  | m | i         | : | p  | s | m |
| 1                                      | 0         | : | 1 | 0 | 2 | 2         | : | 3 | 31 | 2 | 3         | : | 36 | 0 | 0 |
| 2                                      | 0         | : | 3 | 0 | 1 | 1         | : | 4 | 43 | 1 | 3         | : | 48 | 0 | 0 |
| 3                                      | 0         | : | 5 | 0 | 2 | 2         | : | 7 | 21 | 2 | 3         | : | 30 | 0 | 0 |
| 4                                      | 0         | : | 7 | 0 | 1 | 1         | : | 8 | 33 | 1 | 3         | : | 42 | 0 | 0 |

Table D.33: CPP model solution of the 4M3A.s instance.

OBJECTIVE: 14  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 10

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 2 | 3 |

| Positioning, storage, and moving times |           |   |   |   |           |   |   |   |           |    |   |   |   |   |    |   |   |
|----------------------------------------|-----------|---|---|---|-----------|---|---|---|-----------|----|---|---|---|---|----|---|---|
| n                                      | Number: 1 |   |   |   | Number: 2 |   |   |   | Number: 3 |    |   |   |   |   |    |   |   |
| c                                      | i         | p | s | m | i         | p | s | m | i         | p  | s | m |   |   |    |   |   |
| 1                                      | 0         | : | 1 | 0 | 1         | : | 1 | : | 2         | 42 | 1 | : | 3 | : | 45 | 0 | 0 |
| 2                                      | 0         | : | 3 | 0 | 1         | : | 1 | : | 4         | 28 | 1 | : | 3 | : | 33 | 0 | 0 |
| 3                                      | 0         | : | 5 | 0 | 1         | : | 1 | : | 6         | 20 | 1 | : | 3 | : | 27 | 0 | 0 |
| 4                                      | 0         | : | 7 | 0 | 2         | : | 2 | : | 9         | 28 | 2 | : | 3 | : | 39 | 0 | 0 |

Table D.34: CPP model solution of the 4M3B.s instance.



OBJECTIVE: 14  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 10

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 1 | 3 |

| Positioning, storage, and moving times |   |   |   |           |   |   |   |           |   |    |   |   |   |   |    |   |   |
|----------------------------------------|---|---|---|-----------|---|---|---|-----------|---|----|---|---|---|---|----|---|---|
| n   Number: 1                          |   |   |   | Number: 2 |   |   |   | Number: 3 |   |    |   |   |   |   |    |   |   |
| c                                      | i | p | s | m         |   | i | p | s         | m |    | i | p | s | m |    |   |   |
| 1                                      | 0 | : | 1 | 0         | 2 |   | 2 | :         | 3 | 31 | 2 |   | 3 | : | 36 | 0 | 0 |
| 2                                      | 0 | : | 3 | 0         | 1 |   | 1 | :         | 4 | 43 | 1 |   | 3 | : | 48 | 0 | 0 |
| 3                                      | 0 | : | 5 | 0         | 1 |   | 1 | :         | 6 | 35 | 1 |   | 3 | : | 42 | 0 | 0 |
| 4                                      | 0 | : | 7 | 0         | 1 |   | 1 | :         | 8 | 21 | 1 |   | 3 | : | 30 | 0 | 0 |

Table D.35: CPP model solution of the 4M4A.s instance.

OBJECTIVE: 14  
Total number container positionings (sum of xp's) = 4  
Total moving time (sum of xm's) = 10

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 3 |
| 2 | 0 | 2 | 3 |
| 3 | 0 | 1 | 3 |
| 4 | 0 | 1 | 3 |

| n | Number: 1 |   |   |   | Number: 2 |   |   |   | Number: 3 |    |   |   |   |   |    |   |   |
|---|-----------|---|---|---|-----------|---|---|---|-----------|----|---|---|---|---|----|---|---|
| c | i         | p | s | m | i         | p | s | m | i         | p  | s | m |   |   |    |   |   |
| 1 | 0         | : | 1 | 0 | 1         | : | 1 | : | 2         | 51 | 1 | : | 3 | : | 54 | 0 | 0 |
| 2 | 0         | : | 3 | 0 | 2         | : | 2 | : | 5         | 35 | 2 | : | 3 | : | 42 | 0 | 0 |
| 3 | 0         | : | 5 | 0 | 1         | : | 1 | : | 6         | 41 | 1 | : | 3 | : | 48 | 0 | 0 |
| 4 | 0         | : | 7 | 0 | 1         | : | 1 | : | 8         | 27 | 1 | : | 3 | : | 36 | 0 | 0 |

Table D.36: CPP model solution of the 4M4B.s instance.

```

OBJECTIVE: 25
Total number container positionings (sum of xp's) = 7
Total moving time (sum of xm's) = 18

Sequence
c | 1 2 3 4
-----
1 | 0 1 4
2 | 0 2 4
3 | 0 2 1 4
4 | 0 3 4
5 | 0 3 2 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 1 : 3 9 2 | 4 : 14 0 0 |
2 | 0 : 3 0 1 | 2 : 4 11 1 | 4 : 16 0 0 |
3 | 0 : 5 0 1 | 2 : 6 8 1 | 1 : 15 1 2 | 4 : 18 0 0 |
4 | 0 : 7 0 2 | 3 : 9 9 2 | 4 : 20 0 0 |
5 | 0 : 9 0 2 | 3 : 11 3 1 | 2 : 15 6 1 | 4 : 22 0 0 |

```

Table D.37: CPP model solution of the 5S.t instance.

```

OBJECTIVE: 15
Total number container positionings (sum of xp's) = 5
Total moving time (sum of xm's) = 10

Sequence
c | 1 2 3
-----
1 | 0 2 4
2 | 0 2 4
3 | 0 2 4
4 | 0 2 4
5 | 0 2 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 19 1 | 4 : 22 0 0 |
2 | 0 : 3 0 1 | 2 : 4 15 1 | 4 : 20 0 0 |
3 | 0 : 5 0 1 | 2 : 6 11 1 | 4 : 18 0 0 |
4 | 0 : 7 0 1 | 2 : 8 7 1 | 4 : 16 0 0 |
5 | 0 : 9 0 1 | 2 : 10 3 1 | 4 : 14 0 0 |

```

Table D.38: CPP model solution of the 5C.t instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence

| c     | 1 | 2 | 3 |
|-------|---|---|---|
| ----- |   |   |   |
| 1     | 0 | 3 | 4 |
| 2     | 0 | 2 | 4 |
| 3     | 0 | 2 | 4 |
| 4     | 0 | 1 | 4 |
| 5     | 0 | 3 | 4 |

Positioning, storage, and moving times

| n | Number: 1 |   |   |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|---|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1 | 0 | 2 | 3         | : | 3  | 7  | 2 | 4         | : | 12 | 0 | 0 |
| 2 | 0         | : | 3 | 0 | 1 | 2         | : | 4  | 13 | 1 | 4         | : | 18 | 0 | 0 |
| 3 | 0         | : | 5 | 0 | 1 | 2         | : | 6  | 7  | 1 | 4         | : | 14 | 0 | 0 |
| 4 | 0         | : | 7 | 0 | 2 | 1         | : | 9  | 9  | 2 | 4         | : | 20 | 0 | 0 |
| 5 | 0         | : | 9 | 0 | 2 | 3         | : | 11 | 3  | 2 | 4         | : | 16 | 0 | 0 |

Table D.39: CPP model solution of the 5M1A.t instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence

| c     | 1 | 2 | 3 |
|-------|---|---|---|
| ----- |   |   |   |
| 1     | 0 | 3 | 4 |
| 2     | 0 | 2 | 4 |
| 3     | 0 | 1 | 4 |
| 4     | 0 | 2 | 4 |
| 5     | 0 | 3 | 4 |

Positioning, storage, and moving times

| n     | Number: 1 |   |   |   |   | Number: 2 |   |   |    |    | Number: 3 |   |   |   |    |   |   |  |
|-------|-----------|---|---|---|---|-----------|---|---|----|----|-----------|---|---|---|----|---|---|--|
| c     | i         | p | s | m |   | i         | p | s | m  |    | i         | p | s | m |    |   |   |  |
| ----- |           |   |   |   |   |           |   |   |    |    |           |   |   |   |    |   |   |  |
| 1     | 0         | : | 1 | 0 | 2 |           | 3 | : | 3  | 7  | 2         |   | 4 | : | 12 | 0 | 0 |  |
| 2     | 0         | : | 3 | 0 | 1 |           | 2 | : | 4  | 11 | 1         |   | 4 | : | 16 | 0 | 0 |  |
| 3     | 0         | : | 5 | 0 | 2 |           | 1 | : | 7  | 11 | 2         |   | 4 | : | 20 | 0 | 0 |  |
| 4     | 0         | : | 7 | 0 | 1 |           | 2 | : | 8  | 5  | 1         |   | 4 | : | 14 | 0 | 0 |  |
| 5     | 0         | : | 9 | 0 | 2 |           | 3 | : | 11 | 5  | 2         |   | 4 | : | 18 | 0 | 0 |  |

Table D.40: CPP model solution of the 5M1B.t instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 1 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 3 | 4 |

Positioning, storage, and moving times

| n | Number: 1 |   |   |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|---|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1 | 0 | 1 | 2         | : | 2  | 13 | 1 | 4         | : | 16 | 0 | 0 |
| 2 | 0         | : | 3 | 0 | 1 | 2         | : | 4  | 7  | 1 | 4         | : | 12 | 0 | 0 |
| 3 | 0         | : | 5 | 0 | 2 | 1         | : | 7  | 9  | 2 | 4         | : | 18 | 0 | 0 |
| 4 | 0         | : | 7 | 0 | 2 | 1         | : | 9  | 3  | 2 | 4         | : | 14 | 0 | 0 |
| 5 | 0         | : | 9 | 0 | 2 | 3         | : | 11 | 7  | 2 | 4         | : | 20 | 0 | 0 |

Table D.41: CPP model solution of the 5M2A.t instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 3 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 2 | 4 |

| Positioning, storage, and moving times |   |   |   |   |           |   |   |   |   |           |    |   |  |   |   |    |   |   |  |
|----------------------------------------|---|---|---|---|-----------|---|---|---|---|-----------|----|---|--|---|---|----|---|---|--|
| n   Number: 1                          |   |   |   |   | Number: 2 |   |   |   |   | Number: 3 |    |   |  |   |   |    |   |   |  |
| c                                      | i | p | s | m | i         | p | s | m | i | p         | s  | m |  |   |   |    |   |   |  |
| 1                                      |   | 0 | : | 1 | 0         | 1 |   | 2 | : | 2         | 15 | 1 |  | 4 | : | 18 | 0 | 0 |  |
| 2                                      |   | 0 | : | 3 | 0         | 2 |   | 3 | : | 5         | 5  | 2 |  | 4 | : | 12 | 0 | 0 |  |
| 3                                      |   | 0 | : | 5 | 0         | 1 |   | 2 | : | 6         | 9  | 1 |  | 4 | : | 16 | 0 | 0 |  |
| 4                                      |   | 0 | : | 7 | 0         | 2 |   | 1 | : | 9         | 9  | 2 |  | 4 | : | 20 | 0 | 0 |  |
| 5                                      |   | 0 | : | 9 | 0         | 1 |   | 2 | : | 10        | 3  | 1 |  | 4 | : | 14 | 0 | 0 |  |

Table D.42: CPP model solution of the 5M2B.t instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 3 4  
3 | 0 2 4  
4 | 0 3 4  
5 | 0 1 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 11 1 | 4 : 14 0 0 |  
2 | 0 : 3 0 2 | 3 : 5 11 2 | 4 : 18 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 5 1 | 4 : 12 0 0 |  
4 | 0 : 7 0 2 | 3 : 9 5 2 | 4 : 16 0 0 |  
5 | 0 : 9 0 2 | 1 : 11 7 2 | 4 : 20 0 0 |

Table D.43: CPP model solution of the 5M3A.t instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 3 4  
2 | 0 2 4  
3 | 0 1 4  
4 | 0 2 4  
5 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 3 : 3 11 2 | 4 : 16 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 15 1 | 4 : 20 0 0 |  
3 | 0 : 5 0 2 | 1 : 7 3 2 | 4 : 12 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 9 1 | 4 : 18 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 3 1 | 4 : 14 0 0 |

Table D.44: CPP model solution of the 5M3B.t instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 3 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 1 | 4 |

| Positioning, storage, and moving times |           |   |   |   |   |           |   |    |    |   |           |   |    |   |   |
|----------------------------------------|-----------|---|---|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| n                                      | Number: 1 |   |   |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
| c                                      | i         | p | s | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| <hr/>                                  |           |   |   |   |   |           |   |    |    |   |           |   |    |   |   |
| 1                                      | 0         | : | 1 | 0 | 2 | 3         | : | 3  | 9  | 2 | 4         | : | 14 | 0 | 0 |
| 2                                      | 0         | : | 3 | 0 | 1 | 2         | : | 4  | 15 | 1 | 4         | : | 20 | 0 | 0 |
| 3                                      | 0         | : | 5 | 0 | 1 | 2         | : | 6  | 9  | 1 | 4         | : | 16 | 0 | 0 |
| 4                                      | 0         | : | 7 | 0 | 1 | 2         | : | 8  | 3  | 1 | 4         | : | 12 | 0 | 0 |
| 5                                      | 0         | : | 9 | 0 | 2 | 1         | : | 11 | 5  | 2 | 4         | : | 18 | 0 | 0 |

Table D.45: CPP model solution of the 5M4A.t instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 1 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 3 | 4 |

| Positioning, storage, and moving times |   |   |   |   |           |   |   |    |    |           |   |   |    |   |   |
|----------------------------------------|---|---|---|---|-----------|---|---|----|----|-----------|---|---|----|---|---|
| n   Number: 1                          |   |   |   |   | Number: 2 |   |   |    |    | Number: 3 |   |   |    |   |   |
| c                                      | i | p | s | m | i         | p | s | m  | i  | p         | s | m |    |   |   |
| 1                                      | 0 | : | 1 | 0 | 1         | 2 | : | 2  | 13 | 1         | 4 | : | 16 | 0 | 0 |
| 2                                      | 0 | : | 3 | 0 | 1         | 2 | : | 4  | 9  | 1         | 4 | : | 14 | 0 | 0 |
| 3                                      | 0 | : | 5 | 0 | 2         | 1 | : | 7  | 9  | 2         | 4 | : | 18 | 0 | 0 |
| 4                                      | 0 | : | 7 | 0 | 1         | 2 | : | 8  | 3  | 1         | 4 | : | 12 | 0 | 0 |
| 5                                      | 0 | : | 9 | 0 | 2         | 3 | : | 11 | 7  | 2         | 4 | : | 20 | 0 | 0 |

Table D.46: CPP model solution of the 5M4B.t instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence

| c     | 1 | 2 | 3 |
|-------|---|---|---|
| ----- |   |   |   |
| 1     | 0 | 2 | 4 |
| 2     | 0 | 2 | 4 |
| 3     | 0 | 1 | 4 |
| 4     | 0 | 3 | 4 |
| 5     | 0 | 2 | 4 |

Positioning, storage, and moving times

| n     | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |    |    |   |   |   |    |   |   |
|-------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|----|---|---|
| c     | i         | p         | s         | m | i | p | s | m  | i  | p | s | m |    |   |   |
| ----- |           |           |           |   |   |   |   |    |    |   |   |   |    |   |   |
| 1     | 0         | :         | 1         | 0 | 1 | 2 | : | 2  | 15 | 1 | 4 | : | 18 | 0 | 0 |
| 2     | 0         | :         | 3         | 0 | 1 | 2 | : | 4  | 9  | 1 | 4 | : | 14 | 0 | 0 |
| 3     | 0         | :         | 5         | 0 | 2 | 1 | : | 7  | 11 | 2 | 4 | : | 20 | 0 | 0 |
| 4     | 0         | :         | 7         | 0 | 2 | 3 | : | 9  | 5  | 2 | 4 | : | 16 | 0 | 0 |
| 5     | 0         | :         | 9         | 0 | 1 | 2 | : | 10 | 1  | 1 | 4 | : | 12 | 0 | 0 |

Table D.47: CPP model solution of the 5M5A.t instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence

| c     | 1 | 2 | 3 |
|-------|---|---|---|
| ----- |   |   |   |
| 1     | 0 | 2 | 4 |
| 2     | 0 | 3 | 4 |
| 3     | 0 | 2 | 4 |
| 4     | 0 | 3 | 4 |
| 5     | 0 | 2 | 4 |

Positioning, storage, and moving times

| n     | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |    |    |   |   |   |    |   |   |
|-------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|----|---|---|
| c     | i         | p         | s         | m | i | p | s | m  | i  | p | s | m |    |   |   |
| ----- |           |           |           |   |   |   |   |    |    |   |   |   |    |   |   |
| 1     | 0         | :         | 1         | 0 | 1 | 2 | : | 2  | 13 | 1 | 4 | : | 16 | 0 | 0 |
| 2     | 0         | :         | 3         | 0 | 2 | 3 | : | 5  | 13 | 2 | 4 | : | 20 | 0 | 0 |
| 3     | 0         | :         | 5         | 0 | 1 | 2 | : | 6  | 7  | 1 | 4 | : | 14 | 0 | 0 |
| 4     | 0         | :         | 7         | 0 | 2 | 3 | : | 9  | 7  | 2 | 4 | : | 18 | 0 | 0 |
| 5     | 0         | :         | 9         | 0 | 1 | 2 | : | 10 | 1  | 1 | 4 | : | 12 | 0 | 0 |

Table D.48: CPP model solution of the 5M5B.t instance.

```

OBJECTIVE: 25
Total number container positionings (sum of xp's) = 7
Total moving time (sum of xm's) = 18

Sequence
c | 1 2 3 4
-----
1 | 0 1 4
2 | 0 2 3 4
3 | 0 3 4
4 | 0 2 4
5 | 0 2 1 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 1 : 3 37 2 | 4 : 42 0 0 |
2 | 0 : 3 0 1 | 2 : 4 2 1 | 3 : 7 39 2 | 4 : 48 0 0 |
3 | 0 : 5 0 2 | 3 : 7 45 2 | 4 : 54 0 0 |
4 | 0 : 7 0 1 | 2 : 8 51 1 | 4 : 60 0 0 |
5 | 0 : 9 0 1 | 2 : 10 29 1 | 1 : 40 24 2 | 4 : 66 0 0 |

```

Table D.49: CPP model solution of the 5S.s instance.

```

OBJECTIVE: 15
Total number container positionings (sum of xp's) = 5
Total moving time (sum of xm's) = 10

Sequence
c | 1 2 3
-----
1 | 0 2 4
2 | 0 2 4
3 | 0 2 4
4 | 0 2 4
5 | 0 2 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 63 1 | 4 : 66 0 0 |
2 | 0 : 3 0 1 | 2 : 4 55 1 | 4 : 60 0 0 |
3 | 0 : 5 0 1 | 2 : 6 47 1 | 4 : 54 0 0 |
4 | 0 : 7 0 1 | 2 : 8 39 1 | 4 : 48 0 0 |
5 | 0 : 9 0 1 | 2 : 10 31 1 | 4 : 42 0 0 |

```

Table D.50: CPP model solution of the 5C.s instance.



OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence  
c | 1 2 3  
-----  
1 | 0 3 4  
2 | 0 2 4  
3 | 0 2 4  
4 | 0 1 4  
5 | 0 1 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 3 : 3 31 2 | 4 : 36 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 49 1 | 4 : 54 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 35 1 | 4 : 42 0 0 |  
4 | 0 : 7 0 2 | 1 : 9 49 2 | 4 : 60 0 0 |  
5 | 0 : 9 0 2 | 1 : 11 35 2 | 4 : 48 0 0 |

Table D.51: CPP model solution of the 5M1A.s instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence  
c | 1 2 3  
-----  
1 | 0 3 4  
2 | 0 2 4  
3 | 0 1 4  
4 | 0 2 4  
5 | 0 1 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 3 : 3 31 2 | 4 : 36 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 43 1 | 4 : 48 0 0 |  
3 | 0 : 5 0 2 | 1 : 7 51 2 | 4 : 60 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 33 1 | 4 : 42 0 0 |  
5 | 0 : 9 0 2 | 1 : 11 41 2 | 4 : 54 0 0 |

Table D.52: CPP model solution of the 5M1B.s instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 1 4  
4 | 0 1 4  
5 | 0 3 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 45 1 | 4 : 48 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 31 1 | 4 : 36 0 0 |  
3 | 0 : 5 0 2 | 1 : 7 45 2 | 4 : 54 0 0 |  
4 | 0 : 7 0 2 | 1 : 9 31 2 | 4 : 42 0 0 |  
5 | 0 : 9 0 2 | 3 : 11 47 2 | 4 : 60 0 0 |

Table D.53: CPP model solution of the 5M2A.s instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 1 4  
3 | 0 2 4  
4 | 0 3 4  
5 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 51 1 | 4 : 54 0 0 |  
2 | 0 : 3 0 2 | 1 : 5 29 2 | 4 : 36 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 41 1 | 4 : 48 0 0 |  
4 | 0 : 7 0 2 | 3 : 9 49 2 | 4 : 60 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 31 1 | 4 : 42 0 0 |

Table D.54: CPP model solution of the 5M2B.s instance.

OBJECTIVE: 21  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 16

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 3 4  
3 | 0 2 4  
4 | 0 3 4  
5 | 0 1 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 39 1 | 4 : 42 0 0 |  
2 | 0 : 3 0 2 | 3 : 5 47 2 | 4 : 54 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 29 1 | 4 : 36 0 0 |  
4 | 0 : 7 0 2 | 3 : 9 37 2 | 4 : 48 0 0 |  
5 | 0 : 9 0 2 | 1 : 11 47 2 | 4 : 60 0 0 |

Table D.55: CPP model solution of the 5M3A.s instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 1 4  
2 | 0 2 4  
3 | 0 1 4  
4 | 0 2 4  
5 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 1 : 3 43 2 | 4 : 48 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 55 1 | 4 : 60 0 0 |  
3 | 0 : 5 0 2 | 1 : 7 27 2 | 4 : 36 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 45 1 | 4 : 54 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 31 1 | 4 : 42 0 0 |

Table D.56: CPP model solution of the 5M3B.s instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 3 4  
2 | 0 2 4  
3 | 0 2 4  
4 | 0 2 4  
5 | 0 1 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 3 : 3 37 2 | 4 : 42 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 55 1 | 4 : 60 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 41 1 | 4 : 48 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 27 1 | 4 : 36 0 0 |  
5 | 0 : 9 0 2 | 1 : 11 41 2 | 4 : 54 0 0 |

Table D.57: CPP model solution of the 5M4A.s instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 1 4  
4 | 0 2 4  
5 | 0 3 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 45 1 | 4 : 48 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 37 1 | 4 : 42 0 0 |  
3 | 0 : 5 0 2 | 1 : 7 45 2 | 4 : 54 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 27 1 | 4 : 36 0 0 |  
5 | 0 : 9 0 2 | 3 : 11 47 2 | 4 : 60 0 0 |

Table D.58: CPP model solution of the 5M4B.s instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 1 4  
4 | 0 3 4  
5 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 51 1 | 4 : 54 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 37 1 | 4 : 42 0 0 |  
3 | 0 : 5 0 2 | 1 : 7 51 2 | 4 : 60 0 0 |  
4 | 0 : 7 0 2 | 3 : 9 37 2 | 4 : 48 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 25 1 | 4 : 36 0 0 |

Table D.59: CPP model solution of the 5M5A.s instance.

OBJECTIVE: 19  
Total number container positionings (sum of xp's) = 5  
Total moving time (sum of xm's) = 14

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 3 4  
3 | 0 2 4  
4 | 0 1 4  
5 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 45 1 | 4 : 48 0 0 |  
2 | 0 : 3 0 2 | 3 : 5 53 2 | 4 : 60 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 35 1 | 4 : 42 0 0 |  
4 | 0 : 7 0 2 | 1 : 9 43 2 | 4 : 54 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 25 1 | 4 : 36 0 0 |

Table D.60: CPP model solution of the 5M5B.s instance.

## D.2 Medium-scale class instances

```

OBJECTIVE: 31
Total number container positionings (sum of xp's) = 9
Total moving time (sum of xm's) = 22

Sequence
c | 1 2 3 4
-----
1 | 0 2 3 4
2 | 0 2 3 4
3 | 0 3 4
4 | 0 1 4
5 | 0 2 4
6 | 0 2 3 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 5 1 | 3 : 8 8 2 | 4 : 18 0 0 |
2 | 0 : 3 0 1 | 2 : 4 2 1 | 3 : 7 11 2 | 4 : 20 0 0 |
3 | 0 : 5 0 2 | 3 : 7 13 2 | 4 : 22 0 0 |
4 | 0 : 7 0 2 | 1 : 9 13 2 | 4 : 24 0 0 |
5 | 0 : 9 0 1 | 2 : 10 15 1 | 4 : 26 0 0 |
6 | 0 : 11 0 1 | 2 : 12 7 1 | 3 : 20 6 2 | 4 : 28 0 0 |

```

Table D.61: CPP model solution of the 6S.t instance.

```

OBJECTIVE: 18
Total number container positionings (sum of xp's) = 6
Total moving time (sum of xm's) = 12

Sequence
c | 1 2 3
-----
1 | 0 2 4
2 | 0 2 4
3 | 0 2 4
4 | 0 2 4
5 | 0 2 4
6 | 0 2 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 23 1 | 4 : 26 0 0 |
2 | 0 : 3 0 1 | 2 : 4 19 1 | 4 : 24 0 0 |
3 | 0 : 5 0 1 | 2 : 6 15 1 | 4 : 22 0 0 |
4 | 0 : 7 0 1 | 2 : 8 11 1 | 4 : 20 0 0 |
5 | 0 : 9 0 1 | 2 : 10 7 1 | 4 : 18 0 0 |
6 | 0 : 11 0 1 | 2 : 12 3 1 | 4 : 16 0 0 |

```

Table D.62: CPP model solution of the 6C.t instance.

OBJECTIVE: 26  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 20

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 3 | 4 |
| 2 | 0 | 1 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 3 | 4 |

| Positioning, storage, and moving times |   |   |    |   |           |   |   |    |    |           |   |   |      |   |   |
|----------------------------------------|---|---|----|---|-----------|---|---|----|----|-----------|---|---|------|---|---|
| n   Number: 1                          |   |   |    |   | Number: 2 |   |   |    |    | Number: 3 |   |   |      |   |   |
| c                                      | i | p | s  | m | i         | p | s | m  | i  | p         | s | m |      |   |   |
| 1                                      | 0 | : | 1  | 0 | 2         | 3 | : | 3  | 9  | 2         | : | 4 | : 14 | 0 | 0 |
| 2                                      | 0 | : | 3  | 0 | 2         | 1 | : | 5  | 11 | 2         | : | 4 | : 18 | 0 | 0 |
| 3                                      | 0 | : | 5  | 0 | 1         | 2 | : | 6  | 15 | 1         | : | 4 | : 22 | 0 | 0 |
| 4                                      | 0 | : | 7  | 0 | 2         | 1 | : | 9  | 5  | 2         | : | 4 | : 16 | 0 | 0 |
| 5                                      | 0 | : | 9  | 0 | 1         | 2 | : | 10 | 9  | 1         | : | 4 | : 20 | 0 | 0 |
| 6                                      | 0 | : | 11 | 0 | 2         | 3 | : | 13 | 9  | 2         | : | 4 | : 24 | 0 | 0 |

Table D.63: CPP model solution of the 6M1.t instance.

OBJECTIVE: 26  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 20

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 3 | 4 |
| 4 | 0 | 3 | 4 |
| 5 | 0 | 1 | 4 |
| 6 | 0 | 1 | 4 |

| Positioning, storage, and moving times |   |   |    |   |           |   |   |    |    |           |   |   |    |   |   |
|----------------------------------------|---|---|----|---|-----------|---|---|----|----|-----------|---|---|----|---|---|
| n<br>Number: 1                         |   |   |    |   | Number: 2 |   |   |    |    | Number: 3 |   |   |    |   |   |
| c                                      | i | p | s  | m | i         | p | s | m  | i  | p         | s | m |    |   |   |
| 1                                      | 0 | : | 1  | 0 | 1         | 2 | : | 2  | 18 | 1         | 4 | : | 21 | 0 | 0 |
| 2                                      | 0 | : | 3  | 0 | 1         | 2 | : | 4  | 10 | 1         | 4 | : | 15 | 0 | 0 |
| 3                                      | 0 | : | 5  | 0 | 2         | 3 | : | 7  | 14 | 2         | 4 | : | 23 | 0 | 0 |
| 4                                      | 0 | : | 7  | 0 | 2         | 3 | : | 9  | 6  | 2         | 4 | : | 17 | 0 | 0 |
| 5                                      | 0 | : | 9  | 0 | 2         | 1 | : | 11 | 12 | 2         | 4 | : | 25 | 0 | 0 |
| 6                                      | 0 | : | 11 | 0 | 2         | 1 | : | 13 | 4  | 2         | 4 | : | 19 | 0 | 0 |

Table D.64: CPP model solution of the 6M2.t instance.

OBJECTIVE: 22  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 16

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 3 | 4 |
| 6 | 0 | 2 | 4 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s  | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1  | 0 | 1 | 2         | : | 2  | 18 | 1 | 4         | : | 21 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 2         | : | 4  | 12 | 1 | 4         | : | 17 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 2         | : | 6  | 6  | 1 | 4         | : | 13 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 2 | 1         | : | 9  | 12 | 2 | 4         | : | 23 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 2 | 3         | : | 11 | 6  | 2 | 4         | : | 19 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 2         | : | 12 | 2  | 1 | 4         | : | 15 | 0 | 0 |

Table D.65: CPP model solution of the 6M3.t instance.

OBJECTIVE: 24  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 18

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 3 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 3 | 4 |
| 6 | 0 | 1 | 4 |

| Positioning, storage, and moving times |           |   |    |   |   |           |   |    |    |   |           |   |    |   |   |
|----------------------------------------|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| n                                      | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
| c                                      | i         | p | s  | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1                                      | 0         | : | 1  | 0 | 1 | 2         | : | 2  | 21 | 1 | 4         | : | 24 | 0 | 0 |
| 2                                      | 0         | : | 3  | 0 | 1 | 2         | : | 4  | 11 | 1 | 4         | : | 16 | 0 | 0 |
| 3                                      | 0         | : | 5  | 0 | 2 | 3         | : | 7  | 11 | 2 | 4         | : | 20 | 0 | 0 |
| 4                                      | 0         | : | 7  | 0 | 1 | 2         | : | 8  | 5  | 1 | 4         | : | 14 | 0 | 0 |
| 5                                      | 0         | : | 9  | 0 | 2 | 3         | : | 11 | 5  | 2 | 4         | : | 18 | 0 | 0 |
| 6                                      | 0         | : | 11 | 0 | 2 | 1         | : | 13 | 7  | 2 | 4         | : | 22 | 0 | 0 |

Table D.66: CPP model solution of the 6M4.t instance.



OBJECTIVE: 24  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 18

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 1 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 3 | 4 |

Positioning, storage, and moving times

| Number: 1 |   |   |    |   | Number: 2 |   |   |    |    | Number: 3 |   |   |    |   |   |
|-----------|---|---|----|---|-----------|---|---|----|----|-----------|---|---|----|---|---|
| c         | i | p | s  | m | c         | i | p | s  | m  | c         | i | p | s  | m |   |
| 1         | 0 | : | 1  | 0 | 1         | 2 | : | 2  | 17 | 1         | 4 | : | 20 | 0 | 0 |
| 2         | 0 | : | 3  | 0 | 1         | 2 | : | 4  | 11 | 1         | 4 | : | 16 | 0 | 0 |
| 3         | 0 | : | 5  | 0 | 2         | 1 | : | 7  | 13 | 2         | 4 | : | 22 | 0 | 0 |
| 4         | 0 | : | 7  | 0 | 2         | 1 | : | 9  | 7  | 2         | 4 | : | 18 | 0 | 0 |
| 5         | 0 | : | 9  | 0 | 1         | 2 | : | 10 | 3  | 1         | 4 | : | 14 | 0 | 0 |
| 6         | 0 | : | 11 | 0 | 2         | 3 | : | 13 | 9  | 2         | 4 | : | 24 | 0 | 0 |

Table D.67: CPP model solution of the 6M5.t instance.

OBJECTIVE: 22  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 16

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 3 | 4 |
| 6 | 0 | 2 | 4 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s  | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1  | 0 | 1 | 2         | : | 2  | 21 | 1 | 4         | : | 24 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 2         | : | 4  | 15 | 1 | 4         | : | 20 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 2         | : | 6  | 9  | 1 | 4         | : | 16 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 2 | 1         | : | 9  | 11 | 2 | 4         | : | 22 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 2 | 3         | : | 11 | 5  | 2 | 4         | : | 18 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 2         | : | 12 | 1  | 1 | 4         | : | 14 | 0 | 0 |

Table D.68: CPP model solution of the 6M6.t instance.

```

OBJECTIVE: 31
Total number container positionings (sum of xp's) = 9
Total moving time (sum of xm's) = 22

Sequence
c | 1 2 3 4
-----
1 | 0 3 2 4
2 | 0 1 2 4
3 | 0 3 4
4 | 0 2 4
5 | 0 2 1 4
6 | 0 1 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 3 : 3 4 1 | 2 : 8 45 1 | 4 : 54 0 0 |
2 | 0 : 3 0 2 | 1 : 5 2 1 | 2 : 8 51 1 | 4 : 60 0 0 |
3 | 0 : 5 0 2 | 3 : 7 57 2 | 4 : 66 0 0 |
4 | 0 : 7 0 1 | 2 : 8 63 1 | 4 : 72 0 0 |
5 | 0 : 9 0 1 | 2 : 10 2 1 | 1 : 13 63 2 | 4 : 78 0 0 |
6 | 0 : 11 0 2 | 1 : 13 69 2 | 4 : 84 0 0 |

```

Table D.69: CPP model solution of the 6S.s instance.

```

OBJECTIVE: 18
Total number container positionings (sum of xp's) = 6
Total moving time (sum of xm's) = 12

Sequence
c | 1 2 3
-----
1 | 0 2 4
2 | 0 2 4
3 | 0 2 4
4 | 0 2 4
5 | 0 2 4
6 | 0 2 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 75 1 | 4 : 78 0 0 |
2 | 0 : 3 0 1 | 2 : 4 67 1 | 4 : 72 0 0 |
3 | 0 : 5 0 1 | 2 : 6 59 1 | 4 : 66 0 0 |
4 | 0 : 7 0 1 | 2 : 8 51 1 | 4 : 60 0 0 |
5 | 0 : 9 0 1 | 2 : 10 43 1 | 4 : 54 0 0 |
6 | 0 : 11 0 1 | 2 : 12 35 1 | 4 : 48 0 0 |

```

Table D.70: CPP model solution of the 6C.s instance.

```

OBJECTIVE: 27
Total number container positionings (sum of xp's) = 7
Total moving time (sum of xm's) = 20

Sequence
c | 1 2 3 4
-----
1 | 0 3 4
2 | 0 1 4
3 | 0 2 4
4 | 0 1 4
5 | 0 2 4
6 | 0 2 3 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 3 : 3 37 2 | 4 : 42 0 0 |
2 | 0 : 3 0 2 | 1 : 5 47 2 | 4 : 54 0 0 |
3 | 0 : 5 0 1 | 2 : 6 59 1 | 4 : 66 0 0 |
4 | 0 : 7 0 2 | 1 : 9 37 2 | 4 : 48 0 0 |
5 | 0 : 9 0 1 | 2 : 10 49 1 | 4 : 60 0 0 |
6 | 0 : 11 0 1 | 2 : 12 27 1 | 3 : 40 30 2 | 4 : 72 0 0 |

```

Table D.71: CPP model solution of the 6M1.s instance.

```

OBJECTIVE: 26
Total number container positionings (sum of xp's) = 6
Total moving time (sum of xm's) = 20

Sequence
c | 1 2 3
-----
1 | 0 2 4
2 | 0 2 4
3 | 0 3 4
4 | 0 3 4
5 | 0 1 4
6 | 0 1 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 60 1 | 4 : 63 0 0 |
2 | 0 : 3 0 1 | 2 : 4 40 1 | 4 : 45 0 0 |
3 | 0 : 5 0 2 | 3 : 7 60 2 | 4 : 69 0 0 |
4 | 0 : 7 0 2 | 3 : 9 40 2 | 4 : 51 0 0 |
5 | 0 : 9 0 2 | 1 : 11 62 2 | 4 : 75 0 0 |
6 | 0 : 11 0 2 | 1 : 13 42 2 | 4 : 57 0 0 |

```

Table D.72: CPP model solution of the 6M2.s instance.

OBJECTIVE: 24  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 18

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 2 4  
4 | 0 1 4  
5 | 0 3 4  
6 | 0 3 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 60 1 | 4 : 63 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 46 1 | 4 : 51 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 32 1 | 4 : 39 0 0 |  
4 | 0 : 7 0 2 | 1 : 9 58 2 | 4 : 69 0 0 |  
5 | 0 : 9 0 2 | 3 : 11 44 2 | 4 : 57 0 0 |  
6 | 0 : 11 0 2 | 3 : 13 30 2 | 4 : 45 0 0 |

Table D.73: CPP model solution of the 6M3.s instance.

OBJECTIVE: 24  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 18

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 3 4  
4 | 0 2 4  
5 | 0 3 4  
6 | 0 1 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 69 1 | 4 : 72 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 43 1 | 4 : 48 0 0 |  
3 | 0 : 5 0 2 | 3 : 7 51 2 | 4 : 60 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 33 1 | 4 : 42 0 0 |  
5 | 0 : 9 0 2 | 3 : 11 41 2 | 4 : 54 0 0 |  
6 | 0 : 11 0 2 | 1 : 13 51 2 | 4 : 66 0 0 |

Table D.74: CPP model solution of the 6M4.s instance.

OBJECTIVE: 24  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 18

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 1 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 3 | 4 |

| Positioning, storage, and moving times |   |   |    |   |           |   |   |    |    |           |   |   |    |   |   |
|----------------------------------------|---|---|----|---|-----------|---|---|----|----|-----------|---|---|----|---|---|
| n   Number: 1                          |   |   |    |   | Number: 2 |   |   |    |    | Number: 3 |   |   |    |   |   |
| c                                      | i | p | s  | m | i         | p | s | m  | i  | p         | s | m |    |   |   |
| 1                                      | 0 | : | 1  | 0 | 1         | 2 | : | 2  | 57 | 1         | 4 | : | 60 | 0 | 0 |
| 2                                      | 0 | : | 3  | 0 | 1         | 2 | : | 4  | 43 | 1         | 4 | : | 48 | 0 | 0 |
| 3                                      | 0 | : | 5  | 0 | 2         | 1 | : | 7  | 57 | 2         | 4 | : | 66 | 0 | 0 |
| 4                                      | 0 | : | 7  | 0 | 2         | 1 | : | 9  | 43 | 2         | 4 | : | 54 | 0 | 0 |
| 5                                      | 0 | : | 9  | 0 | 1         | 2 | : | 10 | 31 | 1         | 4 | : | 42 | 0 | 0 |
| 6                                      | 0 | : | 11 | 0 | 2         | 3 | : | 13 | 57 | 2         | 4 | : | 72 | 0 | 0 |

Table D.75: CPP model solution of the 6M5.s instance.

OBJECTIVE: 22  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 16

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 1 | 4 |
| 5 | 0 | 3 | 4 |
| 6 | 0 | 2 | 4 |

Positioning, storage, and moving times

| n | Number: 1 |   |   |   |    | Number: 2 |   |   |   |    | Number: 3 |   |   |   |    |   |   |
|---|-----------|---|---|---|----|-----------|---|---|---|----|-----------|---|---|---|----|---|---|
| c | i         | : | p | s | m  | i         | : | p | s | m  | i         | : | p | s | m  |   |   |
| 1 | 1         | : | 0 | : | 1  | 0         | 1 | 2 | : | 2  | 69        | 1 | 4 | : | 72 | 0 | 0 |
| 2 | 1         | : | 0 | : | 3  | 0         | 1 | 2 | : | 4  | 55        | 1 | 4 | : | 60 | 0 | 0 |
| 3 | 1         | : | 0 | : | 5  | 0         | 1 | 2 | : | 6  | 41        | 1 | 4 | : | 48 | 0 | 0 |
| 4 | 1         | : | 0 | : | 7  | 0         | 2 | 1 | : | 9  | 55        | 2 | 4 | : | 66 | 0 | 0 |
| 5 | 1         | : | 0 | : | 9  | 0         | 2 | 3 | : | 11 | 41        | 2 | 4 | : | 54 | 0 | 0 |
| 6 | 1         | : | 0 | : | 11 | 0         | 1 | 2 | : | 12 | 29        | 1 | 4 | : | 42 | 0 | 0 |

Table D.76: CPP model solution of the 6M6.s instance.

```

OBJECTIVE: 42
Total number container positionings (sum of xp's) = 12
Total moving time (sum of xm's) = 30

Sequence
c | 1 2 3 4
-----
1 | 0 1 3 5
2 | 0 1 3 5
3 | 0 3 5
4 | 0 1 5
5 | 0 4 5
6 | 0 2 5
7 | 0 1 3 5
8 | 0 4 3 5

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 5 1 | 3 : 8 16 1 | 5 : 25 0 0 |
2 | 0 : 3 0 1 | 1 : 4 2 1 | 3 : 7 19 1 | 5 : 27 0 0 |
3 | 0 : 5 0 2 | 3 : 7 21 1 | 5 : 29 0 0 |
4 | 0 : 7 0 1 | 1 : 8 21 2 | 5 : 31 0 0 |
5 | 0 : 9 0 3 | 4 : 12 19 2 | 5 : 33 0 0 |
6 | 0 : 11 0 2 | 2 : 13 19 3 | 5 : 35 0 0 |
7 | 0 : 13 0 1 | 1 : 14 13 1 | 3 : 28 8 1 | 5 : 37 0 0 |
8 | 0 : 15 0 3 | 4 : 18 9 1 | 3 : 28 10 1 | 5 : 39 0 0 |

```

Table D.77: CPP model solution of the 8S.t instance.

```

OBJECTIVE: 32
Total number container positionings (sum of xp's) = 8
Total moving time (sum of xm's) = 24

Sequence
c | 1 2 3
-----
1 | 0 1 5
2 | 0 1 5
3 | 0 1 5
4 | 0 1 5
5 | 0 1 5
6 | 0 1 5
7 | 0 1 5
8 | 0 1 5

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 30 2 | 5 : 34 0 0 |
2 | 0 : 3 0 1 | 1 : 4 26 2 | 5 : 32 0 0 |
3 | 0 : 5 0 1 | 1 : 6 22 2 | 5 : 30 0 0 |
4 | 0 : 7 0 1 | 1 : 8 18 2 | 5 : 28 0 0 |
5 | 0 : 9 0 1 | 1 : 10 14 2 | 5 : 26 0 0 |
6 | 0 : 11 0 1 | 1 : 12 10 2 | 5 : 24 0 0 |
7 | 0 : 13 0 1 | 1 : 14 6 2 | 5 : 22 0 0 |
8 | 0 : 15 0 1 | 1 : 16 2 2 | 5 : 20 0 0 |

```

Table D.78: CPP model solution of the 8C.t instance.

OBJECTIVE: 37  
Total number container positionings (sum of xp's) = 9  
Total moving time (sum of xm's) = 28

Sequence

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 4 | 5 |   |
| 2 | 0 | 1 | 3 | 5 |
| 3 | 0 | 3 | 5 |   |
| 4 | 0 | 2 | 5 |   |
| 5 | 0 | 3 | 5 |   |
| 6 | 0 | 1 | 5 |   |
| 7 | 0 | 3 | 5 |   |
| 8 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n | Number: 1  | Number: 2   | Number: 3   | Number: 4  |
|---|------------|-------------|-------------|------------|
| c | i : p s m  | i : p s m   | i : p s m   | i : p s m  |
| 1 | 0 : 1 0 3  | 4 : 4 16 2  | 5 : 22 0 0  |            |
| 2 | 0 : 3 0 1  | 1 : 4 6 1   | 3 : 11 16 1 | 5 : 28 0 0 |
| 3 | 0 : 5 0 2  | 3 : 7 26 1  | 5 : 34 0 0  |            |
| 4 | 0 : 7 0 2  | 2 : 9 12 3  | 5 : 24 0 0  |            |
| 5 | 0 : 9 0 2  | 3 : 11 18 1 | 5 : 30 0 0  |            |
| 6 | 0 : 11 0 1 | 1 : 12 22 2 | 5 : 36 0 0  |            |
| 7 | 0 : 13 0 2 | 3 : 15 10 1 | 5 : 26 0 0  |            |
| 8 | 0 : 15 0 1 | 1 : 16 14 2 | 5 : 32 0 0  |            |

Table D.79: CPP model solution of the 8M1.t instance.

OBJECTIVE: 38  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 28

Sequence

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 |   |
| 2 | 0 | 2 | 5 |   |
| 3 | 0 | 1 | 5 |   |
| 4 | 0 | 3 | 5 |   |
| 5 | 0 | 1 | 3 | 5 |
| 6 | 0 | 4 | 5 |   |
| 7 | 0 | 1 | 3 | 5 |
| 8 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n | Number: 1  | Number: 2   | Number: 3  | Number: 4  |
|---|------------|-------------|------------|------------|
| c | i : p s m  | i : p s m   | i : p s m  | i : p s m  |
| 1 | 0 : 1 0 2  | 3 : 3 26 1  | 5 : 30 0 0 |            |
| 2 | 0 : 3 0 2  | 2 : 5 14 3  | 5 : 22 0 0 |            |
| 3 | 0 : 5 0 1  | 1 : 6 24 2  | 5 : 32 0 0 |            |
| 4 | 0 : 7 0 2  | 3 : 9 14 1  | 5 : 24 0 0 |            |
| 5 | 0 : 9 0 1  | 1 : 10 19 1 | 3 : 30 3 1 | 5 : 34 0 0 |
| 6 | 0 : 11 0 3 | 4 : 14 10 2 | 5 : 26 0 0 |            |
| 7 | 0 : 13 0 1 | 1 : 14 14 1 | 3 : 29 6 1 | 5 : 36 0 0 |
| 8 | 0 : 15 0 1 | 1 : 16 10 2 | 5 : 28 0 0 |            |

Table D.80: CPP model solution of the 8M2.t instance.

```

OBJECTIVE: 35
Total number container positionings (sum of xp's) = 9
Total moving time (sum of xm's) = 26

Sequence
c | 1 2 3 4
-----
1 | 0 3 5
2 | 0 1 5
3 | 0 3 5
4 | 0 1 5
5 | 0 1 5
6 | 0 1 3 5
7 | 0 1 5
8 | 0 2 5

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 3 : 3 19 1 | 5 : 23 0 0 |
2 | 0 : 3 0 1 | 1 : 4 25 2 | 5 : 31 0 0 |
3 | 0 : 5 0 2 | 3 : 7 13 1 | 5 : 21 0 0 |
4 | 0 : 7 0 1 | 1 : 8 19 2 | 5 : 29 0 0 |
5 | 0 : 9 0 1 | 1 : 10 15 2 | 5 : 27 0 0 |
6 | 0 : 11 0 1 | 1 : 12 12 1 | 3 : 25 9 1 | 5 : 35 0 0 |
7 | 0 : 13 0 1 | 1 : 14 9 2 | 5 : 25 0 0 |
8 | 0 : 15 0 2 | 2 : 17 13 3 | 5 : 33 0 0 |

```

Table D.81: CPP model solution of the 8M3.t instance.

```

OBJECTIVE: 35
Total number container positionings (sum of xp's) = 9
Total moving time (sum of xm's) = 26

Sequence
c | 1 2 3 4
-----
1 | 0 3 5
2 | 0 1 3 5
3 | 0 3 5
4 | 0 4 5
5 | 0 3 5
6 | 0 1 5
7 | 0 3 5
8 | 0 1 5

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 2 | 3 : 3 26 1 | 5 : 30 0 0 |
2 | 0 : 3 0 1 | 1 : 4 6 1 | 3 : 11 10 1 | 5 : 22 0 0 |
3 | 0 : 5 0 2 | 3 : 7 18 1 | 5 : 26 0 0 |
4 | 0 : 7 0 3 | 4 : 10 6 2 | 5 : 18 0 0 |
5 | 0 : 9 0 2 | 3 : 11 12 1 | 5 : 24 0 0 |
6 | 0 : 11 0 1 | 1 : 12 18 2 | 5 : 32 0 0 |
7 | 0 : 13 0 2 | 3 : 15 4 1 | 5 : 20 0 0 |
8 | 0 : 15 0 1 | 1 : 16 10 2 | 5 : 28 0 0 |

```

Table D.82: CPP model solution of the 8M4.t instance.



OBJECTIVE: 33  
 Total number container positionings (sum of xp's) = 9  
 Total moving time (sum of xm's) = 24

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 |   |
| 2 | 0 | 1 | 5 |   |
| 3 | 0 | 3 | 5 |   |
| 4 | 0 | 1 | 5 |   |
| 5 | 0 | 3 | 5 |   |
| 6 | 0 | 1 | 5 |   |
| 7 | 0 | 1 | 3 | 5 |
| 8 | 0 | 1 | 5 |   |

Positioning, storage, and moving times  

| n | Number: 1  | Number: 2   | Number: 3  | Number: 4  |
|---|------------|-------------|------------|------------|
| c | i : p s m  | i : p s m   | i : p s m  | i : p s m  |
| 1 | 0 : 1 0 2  | 3 : 3 28 1  | 5 : 32 0 0 |            |
| 2 | 0 : 3 0 1  | 1 : 4 22 2  | 5 : 28 0 0 |            |
| 3 | 0 : 5 0 2  | 3 : 7 12 1  | 5 : 20 0 0 |            |
| 4 | 0 : 7 0 1  | 1 : 8 16 2  | 5 : 26 0 0 |            |
| 5 | 0 : 9 0 2  | 3 : 11 6 1  | 5 : 18 0 0 |            |
| 6 | 0 : 11 0 1 | 1 : 12 10 2 | 5 : 24 0 0 |            |
| 7 | 0 : 13 0 1 | 1 : 14 7 1  | 3 : 22 7 1 | 5 : 30 0 0 |
| 8 | 0 : 15 0 1 | 1 : 16 4 2  | 5 : 22 0 0 |            |

Table D.83: CPP model solution of the 8M5.t instance.

OBJECTIVE: 35  
 Total number container positionings (sum of xp's) = 11  
 Total moving time (sum of xm's) = 24

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 5 |   |
| 2 | 0 | 1 | 3 | 5 |
| 3 | 0 | 3 | 5 |   |
| 4 | 0 | 1 | 5 |   |
| 5 | 0 | 1 | 3 | 5 |
| 6 | 0 | 3 | 5 |   |
| 7 | 0 | 1 | 5 |   |
| 8 | 0 | 1 | 3 | 5 |

Positioning, storage, and moving times  

| n | Number: 1  | Number: 2   | Number: 3   | Number: 4  |
|---|------------|-------------|-------------|------------|
| c | i : p s m  | i : p s m   | i : p s m   | i : p s m  |
| 1 | 0 : 1 0 1  | 1 : 2 23 2  | 5 : 27 0 0  |            |
| 2 | 0 : 3 0 1  | 1 : 4 20 1  | 3 : 25 3 1  | 5 : 29 0 0 |
| 3 | 0 : 5 0 2  | 3 : 7 13 1  | 5 : 21 0 0  |            |
| 4 | 0 : 7 0 1  | 1 : 8 15 2  | 5 : 25 0 0  |            |
| 5 | 0 : 9 0 1  | 1 : 10 12 1 | 3 : 23 7 1  | 5 : 31 0 0 |
| 6 | 0 : 11 0 2 | 3 : 13 5 1  | 5 : 19 0 0  |            |
| 7 | 0 : 13 0 1 | 1 : 14 7 2  | 5 : 23 0 0  |            |
| 8 | 0 : 15 0 1 | 1 : 16 4 1  | 3 : 21 11 1 | 5 : 33 0 0 |

Table D.84: CPP model solution of the 8M6.t instance.

OBJECTIVE: 34  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 26

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 3 | 5 |
| 5 | 0 | 3 | 5 |
| 6 | 0 | 3 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 4 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p  | s | m |
| 1 | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 24 | 2 | 5         | : | 28 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 1         | : | 4  | 18 | 2 | 5         | : | 24 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 12 | 2 | 5         | : | 20 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 2 | 3         | : | 9  | 20 | 1 | 5         | : | 30 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 2 | 3         | : | 11 | 14 | 1 | 5         | : | 26 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 2 | 3         | : | 13 | 8  | 1 | 5         | : | 22 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 2  | 2 | 5         | : | 18 | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 3 | 4         | : | 18 | 12 | 2 | 5         | : | 32 | 0 | 0 |

Table D.85: CPP model solution of the 8M7.t instance.

OBJECTIVE: 34  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 26

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 4 | 5 |
| 3 | 0 | 3 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 3 | 5 |
| 7 | 0 | 3 | 5 |
| 8 | 0 | 3 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   | Number: 2 |   |   |    | Number: 3 |   |   |   |    |   |   |
|---|-----------|---|----|---|-----------|---|---|----|-----------|---|---|---|----|---|---|
|   | c         | i | p  | s | m         | i | p | s  | m         | i | p | s | m  |   |   |
| 1 | 0         | : | 1  | 0 | 1         | 1 | : | 2  | 25        | 2 | 5 | : | 29 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 3         | 4 | : | 6  | 15        | 2 | 5 | : | 23 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 2         | 3 | : | 7  | 25        | 1 | 5 | : | 33 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1         | 1 | : | 8  | 17        | 2 | 5 | : | 27 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1         | 1 | : | 10 | 9         | 2 | 5 | : | 21 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 2         | 3 | : | 13 | 17        | 1 | 5 | : | 31 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 2         | 3 | : | 15 | 9         | 1 | 5 | : | 25 | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 2         | 3 | : | 17 | 1         | 1 | 5 | : | 19 | 0 | 0 |

Table D.86: CPP model solution of the 8M8.t instance.

OBJECTIVE: 42  
 Total number container positionings (sum of xp's) = 12  
 Total moving time (sum of xm's) = 30

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 5 |   |
| 2 | 0 | 1 | 3 | 5 |
| 3 | 0 | 1 | 3 | 5 |
| 4 | 0 | 4 | 5 |   |
| 5 | 0 | 1 | 3 | 5 |
| 6 | 0 | 1 | 2 | 5 |
| 7 | 0 | 3 | 5 |   |
| 8 | 0 | 2 | 5 |   |

Positioning, storage, and moving times  

| n | Number: 1  | Number: 2   | Number: 3   | Number: 4   |
|---|------------|-------------|-------------|-------------|
| c | i : p s m  | i : p s m   | i : p s m   | i : p s m   |
| 1 | 0 : 1 0 1  | 1 : 2 71 2  | 5 : 75 0 0  |             |
| 2 | 0 : 3 0 1  | 1 : 4 15 1  | 3 : 20 60 1 | 5 : 81 0 0  |
| 3 | 0 : 5 0 1  | 1 : 6 12 1  | 3 : 19 67 1 | 5 : 87 0 0  |
| 4 | 0 : 7 0 3  | 4 : 10 81 2 | 5 : 93 0 0  |             |
| 5 | 0 : 9 0 1  | 1 : 10 7 1  | 3 : 18 80 1 | 5 : 99 0 0  |
| 6 | 0 : 11 0 1 | 1 : 12 4 1  | 2 : 17 85 3 | 5 : 105 0 0 |
| 7 | 0 : 13 0 2 | 3 : 15 95 1 | 5 : 111 0 0 |             |
| 8 | 0 : 15 0 2 | 2 : 17 97 3 | 5 : 117 0 0 |             |

Table D.87: CPP model solution of the 8S.s instance.

OBJECTIVE: 32  
 Total number container positionings (sum of xp's) = 8  
 Total moving time (sum of xm's) = 24

Sequence  

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times  

| n | Number: 1  | Number: 2   | Number: 3   |
|---|------------|-------------|-------------|
| c | i : p s m  | i : p s m   | i : p s m   |
| 1 | 0 : 1 0 1  | 1 : 2 98 2  | 5 : 102 0 0 |
| 2 | 0 : 3 0 1  | 1 : 4 90 2  | 5 : 96 0 0  |
| 3 | 0 : 5 0 1  | 1 : 6 82 2  | 5 : 90 0 0  |
| 4 | 0 : 7 0 1  | 1 : 8 74 2  | 5 : 84 0 0  |
| 5 | 0 : 9 0 1  | 1 : 10 66 2 | 5 : 78 0 0  |
| 6 | 0 : 11 0 1 | 1 : 12 58 2 | 5 : 72 0 0  |
| 7 | 0 : 13 0 1 | 1 : 14 50 2 | 5 : 66 0 0  |
| 8 | 0 : 15 0 1 | 1 : 16 42 2 | 5 : 60 0 0  |

Table D.88: CPP model solution of the 8C.s instance.

OBJECTIVE: 38  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 30

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 4 | 5 |
| 2 | 0 | 2 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 2 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 3 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 3 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |   |   |    | Number: 2 |   |   |   |    | Number: 3 |   |    |   |   |   |     |   |   |
|---|-----------|---|---|---|----|-----------|---|---|---|----|-----------|---|----|---|---|---|-----|---|---|
|   | c         | i | : | p | s  | m         | i | : | p | s  | m         | i | :  | p | s | m |     |   |   |
| 1 | 1         | : | 0 | : | 1  | 0         | 3 | 4 | : | 4  | :         | 6 | 60 | 2 | 5 | : | 66  | 0 | 0 |
| 2 | 1         | : | 0 | : | 3  | 0         | 2 | 2 | : | 5  | :         | 7 | 63 | 2 | 5 | : | 84  | 0 | 0 |
| 3 | 1         | : | 0 | : | 5  | 0         | 1 | 1 | : | 6  | :         | 9 | 64 | 2 | 5 | : | 102 | 0 | 0 |
| 4 | 1         | : | 0 | : | 7  | 0         | 2 | 2 | : | 9  | :         | 6 | 60 | 3 | 5 | : | 72  | 0 | 0 |
| 5 | 1         | : | 0 | : | 9  | 0         | 1 | 1 | : | 10 | :         | 7 | 78 | 2 | 5 | : | 90  | 0 | 0 |
| 6 | 1         | : | 0 | : | 11 | 0         | 2 | 3 | : | 13 | :         | 9 | 94 | 1 | 5 | : | 108 | 0 | 0 |
| 7 | 1         | : | 0 | : | 13 | 0         | 1 | 1 | : | 14 | :         | 6 | 62 | 2 | 5 | : | 78  | 0 | 0 |
| 8 | 1         | : | 0 | : | 15 | 0         | 2 | 3 | : | 17 | :         | 7 | 78 | 1 | 5 | : | 96  | 0 | 0 |

Table D.89: CPP model solution of the 8M1.s instance.

OBJECTIVE: 38  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 28

Sequence

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 |   |
| 2 | 0 | 3 | 5 |   |
| 3 | 0 | 1 | 5 |   |
| 4 | 0 | 2 | 5 |   |
| 5 | 0 | 1 | 3 | 5 |
| 6 | 0 | 4 | 5 |   |
| 7 | 0 | 1 | 3 | 5 |
| 8 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |    |   | Number: 4 |   |     |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|----|---|-----------|---|-----|---|---|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p  | s  | m | i         | : | p   | s | m |
| 1 | 0         | : | 1  | 0 | 2 | 3         | : | 3  | 86 | 1 | 5         | : | 90 | 0  | 0 |           |   |     |   |   |
| 2 | 0         | : | 3  | 0 | 2 | 3         | : | 5  | 60 | 1 | 5         | : | 66 | 0  | 0 |           |   |     |   |   |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 88 | 2 | 5         | : | 96 | 0  | 0 |           |   |     |   |   |
| 4 | 0         | : | 7  | 0 | 2 | 2         | : | 9  | 60 | 3 | 5         | : | 72 | 0  | 0 |           |   |     |   |   |
| 5 | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 83 | 1 | 3         | : | 94 | 7  | 1 | 5         | : | 102 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 3 | 4         | : | 14 | 62 | 2 | 5         | : | 78 | 0  | 0 |           |   |     |   |   |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 78 | 1 | 3         | : | 93 | 14 | 1 | 5         | : | 108 | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1 | 1         | : | 16 | 66 | 2 | 5         | : | 84 | 0  | 0 |           |   |     |   |   |

Table D.90: CPP model solution of the 8M2.s instance.

OBJECTIVE: 35  
Total number container positionings (sum of xp's) = 9  
Total moving time (sum of xm's) = 26

Sequence

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 |   |
| 2 | 0 | 1 | 5 |   |
| 3 | 0 | 3 | 5 |   |
| 4 | 0 | 1 | 5 |   |
| 5 | 0 | 1 | 5 |   |
| 6 | 0 | 2 | 5 |   |
| 7 | 0 | 1 | 5 |   |
| 8 | 0 | 1 | 3 | 5 |

Positioning, storage, and moving times

| n | Number: 1 | Number: 2  | Number: 3  | Number: 4 |
|---|-----------|------------|------------|-----------|
| c | i: p s m  | i: p s m   | i: p s m   | i: p s m  |
| 1 | 0: 1 0 2  | 3: 3 65 1  | 5: 69 0 0  |           |
| 2 | 0: 3 0 1  | 1: 4 87 2  | 5: 93 0 0  |           |
| 3 | 0: 5 0 2  | 3: 7 55 1  | 5: 63 0 0  |           |
| 4 | 0: 7 0 1  | 1: 8 77 2  | 5: 87 0 0  |           |
| 5 | 0: 9 0 1  | 1: 10 69 2 | 5: 81 0 0  |           |
| 6 | 0: 11 0 2 | 2: 13 89 3 | 5: 105 0 0 |           |
| 7 | 0: 13 0 1 | 1: 14 59 2 | 5: 75 0 0  |           |
| 8 | 0: 15 0 1 | 1: 16 51 1 | 3: 68 30 1 | 5: 99 0 0 |

Table D.91: CPP model solution of the 8M3.s instance.

OBJECTIVE: 35  
Total number container positionings (sum of xp's) = 9  
Total moving time (sum of xm's) = 26

Sequence

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 |   |
| 2 | 0 | 1 | 3 | 5 |
| 3 | 0 | 3 | 5 |   |
| 4 | 0 | 4 | 5 |   |
| 5 | 0 | 3 | 5 |   |
| 6 | 0 | 1 | 5 |   |
| 7 | 0 | 3 | 5 |   |
| 8 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n | Number: 1 | Number: 2  | Number: 3  | Number: 4 |
|---|-----------|------------|------------|-----------|
| c | i: p s m  | i: p s m   | i: p s m   | i: p s m  |
| 1 | 0: 1 0 2  | 3: 3 86 1  | 5: 90 0 0  |           |
| 2 | 0: 3 0 1  | 1: 4 6 1   | 3: 11 54 1 | 5: 66 0 0 |
| 3 | 0: 5 0 2  | 3: 7 70 1  | 5: 78 0 0  |           |
| 4 | 0: 7 0 3  | 4: 10 42 2 | 5: 54 0 0  |           |
| 5 | 0: 9 0 2  | 3: 11 60 1 | 5: 72 0 0  |           |
| 6 | 0: 11 0 1 | 1: 12 82 2 | 5: 96 0 0  |           |
| 7 | 0: 13 0 2 | 3: 15 44 1 | 5: 60 0 0  |           |
| 8 | 0: 15 0 1 | 1: 16 66 2 | 5: 84 0 0  |           |

Table D.92: CPP model solution of the 8M4.s instance.

OBJECTIVE: 33  
 Total number container positionings (sum of xp's) = 9  
 Total moving time (sum of xm's) = 24

Sequence  

| c | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 3 | 5 |   |
| 2 | 0 | 1 | 5 |   |
| 3 | 0 | 3 | 5 |   |
| 4 | 0 | 1 | 5 |   |
| 5 | 0 | 3 | 5 |   |
| 6 | 0 | 1 | 5 |   |
| 7 | 0 | 1 | 3 | 5 |
| 8 | 0 | 1 | 5 |   |

Positioning, storage, and moving times  

| n | Number: 1  | Number: 2   | Number: 3   | Number: 4  |
|---|------------|-------------|-------------|------------|
| c | i : p s m  | i : p s m   | i : p s m   | i : p s m  |
| 1 | 0 : 1 0 2  | 3 : 3 92 1  | 5 : 96 0 0  |            |
| 2 | 0 : 3 0 1  | 1 : 4 78 2  | 5 : 84 0 0  |            |
| 3 | 0 : 5 0 2  | 3 : 7 52 1  | 5 : 60 0 0  |            |
| 4 | 0 : 7 0 1  | 1 : 8 68 2  | 5 : 78 0 0  |            |
| 5 | 0 : 9 0 2  | 3 : 11 42 1 | 5 : 54 0 0  |            |
| 6 | 0 : 11 0 1 | 1 : 12 58 2 | 5 : 72 0 0  |            |
| 7 | 0 : 13 0 1 | 1 : 14 51 1 | 3 : 66 23 1 | 5 : 90 0 0 |
| 8 | 0 : 15 0 1 | 1 : 16 48 2 | 5 : 66 0 0  |            |

Table D.93: CPP model solution of the 8M5.s instance.

OBJECTIVE: 36  
 Total number container positionings (sum of xp's) = 8  
 Total moving time (sum of xm's) = 28

Sequence  

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 3 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 3 | 5 |
| 5 | 0 | 4 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 3 | 5 |
| 8 | 0 | 2 | 5 |

Positioning, storage, and moving times  

| n | Number: 1  | Number: 2   | Number: 3  |
|---|------------|-------------|------------|
| c | i : p s m  | i : p s m   | i : p s m  |
| 1 | 0 : 1 0 1  | 1 : 2 77 2  | 5 : 81 0 0 |
| 2 | 0 : 3 0 2  | 3 : 5 81 1  | 5 : 87 0 0 |
| 3 | 0 : 5 0 1  | 1 : 6 55 2  | 5 : 63 0 0 |
| 4 | 0 : 7 0 2  | 3 : 9 65 1  | 5 : 75 0 0 |
| 5 | 0 : 9 0 3  | 4 : 12 79 2 | 5 : 93 0 0 |
| 6 | 0 : 11 0 1 | 1 : 12 43 2 | 5 : 57 0 0 |
| 7 | 0 : 13 0 2 | 3 : 15 53 1 | 5 : 69 0 0 |
| 8 | 0 : 15 0 2 | 2 : 17 79 3 | 5 : 99 0 0 |

Table D.94: CPP model solution of the 8M6.s instance.

OBJECTIVE: 34  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 26

Sequence  
c | 1 2 3  
-----  
1 | 0 3 5  
2 | 0 3 5  
3 | 0 3 5  
4 | 0 1 5  
5 | 0 1 5  
6 | 0 1 5  
7 | 0 1 5  
8 | 0 2 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 3 : 3 80 1 | 5 : 84 0 0 |  
2 | 0 : 3 0 2 | 3 : 5 66 1 | 5 : 72 0 0 |  
3 | 0 : 5 0 2 | 3 : 7 52 1 | 5 : 60 0 0 |  
4 | 0 : 7 0 1 | 1 : 8 80 2 | 5 : 90 0 0 |  
5 | 0 : 9 0 1 | 1 : 10 66 2 | 5 : 78 0 0 |  
6 | 0 : 11 0 1 | 1 : 12 52 2 | 5 : 66 0 0 |  
7 | 0 : 13 0 1 | 1 : 14 38 2 | 5 : 54 0 0 |  
8 | 0 : 15 0 2 | 2 : 17 76 3 | 5 : 96 0 0 |

Table D.95: CPP model solution of the 8M7.s instance.

OBJECTIVE: 34  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 26

Sequence  
c | 1 2 3  
-----  
1 | 0 1 5  
2 | 0 1 5  
3 | 0 3 5  
4 | 0 3 5  
5 | 0 1 5  
6 | 0 2 5  
7 | 0 3 5  
8 | 0 3 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 1 : 2 83 2 | 5 : 87 0 0 |  
2 | 0 : 3 0 1 | 1 : 4 63 2 | 5 : 69 0 0 |  
3 | 0 : 5 0 2 | 3 : 7 91 1 | 5 : 99 0 0 |  
4 | 0 : 7 0 2 | 3 : 9 71 1 | 5 : 81 0 0 |  
5 | 0 : 9 0 1 | 1 : 10 51 2 | 5 : 63 0 0 |  
6 | 0 : 11 0 2 | 2 : 13 77 3 | 5 : 93 0 0 |  
7 | 0 : 13 0 2 | 3 : 15 59 1 | 5 : 75 0 0 |  
8 | 0 : 15 0 2 | 3 : 17 39 1 | 5 : 57 0 0 |

Table D.96: CPP model solution of the 8M8.s instance.

OBJECTIVE: 52  
 Total number container positionings (sum of xp's) = 14  
 Total moving time (sum of xm's) = 38

Sequence  

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 4 | 5 |   |
| 2  | 0 | 2 | 5 |   |
| 3  | 0 | 1 | 5 |   |
| 4  | 0 | 1 | 3 | 5 |
| 5  | 0 | 1 | 3 | 5 |
| 6  | 0 | 1 | 3 | 5 |
| 7  | 0 | 3 | 5 |   |
| 8  | 0 | 1 | 2 | 5 |
| 9  | 0 | 4 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times  

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |    |   |   |   |   |    |    |   |  |   |   |    |   |   |  |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|---|----|----|---|--|---|---|----|---|---|--|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p  | s | m | i | p | s  | m  |   |  |   |   |    |   |   |  |
| 1  | 0         | :         | 1         | 0         | 3 |   | 4 | : | 4  | 23 | 2 |   | 5 | : | 29 | 0  | 0 |  |   |   |    |   |   |  |
| 2  | 0         | :         | 5         | 0         | 2 |   | 2 | : | 7  | 23 | 3 |   | 5 | : | 33 | 0  | 0 |  |   |   |    |   |   |  |
| 3  | 0         | :         | 9         | 0         | 1 |   | 1 | : | 10 | 25 | 2 |   | 5 | : | 37 | 0  | 0 |  |   |   |    |   |   |  |
| 4  | 0         | :         | 13        | 0         | 1 |   | 1 | : | 14 | 14 | 1 |   | 3 | : | 29 | 11 | 1 |  | 5 | : | 41 | 0 | 0 |  |
| 5  | 0         | :         | 17        | 0         | 1 |   | 1 | : | 18 | 9  | 1 |   | 3 | : | 28 | 16 | 1 |  | 5 | : | 45 | 0 | 0 |  |
| 6  | 0         | :         | 21        | 0         | 1 |   | 1 | : | 22 | 4  | 1 |   | 3 | : | 27 | 21 | 1 |  | 5 | : | 49 | 0 | 0 |  |
| 7  | 0         | :         | 25        | 0         | 2 |   | 3 | : | 27 | 25 | 1 |   | 5 | : | 53 | 0  | 0 |  |   |   |    |   |   |  |
| 8  | 0         | :         | 29        | 0         | 1 |   | 1 | : | 30 | 0  | 1 |   | 2 | : | 31 | 23 | 3 |  | 5 | : | 57 | 0 | 0 |  |
| 9  | 0         | :         | 33        | 0         | 3 |   | 4 | : | 36 | 23 | 2 |   | 5 | : | 61 | 0  | 0 |  |   |   |    |   |   |  |
| 10 | 0         | :         | 37        | 0         | 1 |   | 1 | : | 38 | 25 | 2 |   | 5 | : | 65 | 0  | 0 |  |   |   |    |   |   |  |

Table D.97: CPP model solution of the 10S.t instance.

OBJECTIVE: 40  
 Total number container positionings (sum of xp's) = 10  
 Total moving time (sum of xm's) = 30

Sequence  

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times  

| n  | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |   |    |    |   |   |   |   |    |   |   |  |
|----|-----------|-----------|-----------|---|---|---|---|---|----|----|---|---|---|---|----|---|---|--|
| c  | i         | p         | s         | m | i | p | s | m | i  | p  | s | m |   |   |    |   |   |  |
| 1  | 0         | :         | 1         | 0 | 1 |   | 1 | : | 2  | 74 | 2 |   | 5 | : | 78 | 0 | 0 |  |
| 2  | 0         | :         | 5         | 0 | 1 |   | 1 | : | 6  | 66 | 2 |   | 5 | : | 74 | 0 | 0 |  |
| 3  | 0         | :         | 9         | 0 | 1 |   | 1 | : | 10 | 58 | 2 |   | 5 | : | 70 | 0 | 0 |  |
| 4  | 0         | :         | 13        | 0 | 1 |   | 1 | : | 14 | 50 | 2 |   | 5 | : | 66 | 0 | 0 |  |
| 5  | 0         | :         | 17        | 0 | 1 |   | 1 | : | 18 | 42 | 2 |   | 5 | : | 62 | 0 | 0 |  |
| 6  | 0         | :         | 21        | 0 | 1 |   | 1 | : | 22 | 34 | 2 |   | 5 | : | 58 | 0 | 0 |  |
| 7  | 0         | :         | 25        | 0 | 1 |   | 1 | : | 26 | 26 | 2 |   | 5 | : | 54 | 0 | 0 |  |
| 8  | 0         | :         | 29        | 0 | 1 |   | 1 | : | 30 | 18 | 2 |   | 5 | : | 50 | 0 | 0 |  |
| 9  | 0         | :         | 33        | 0 | 1 |   | 1 | : | 34 | 10 | 2 |   | 5 | : | 46 | 0 | 0 |  |
| 10 | 0         | :         | 37        | 0 | 1 |   | 1 | : | 38 | 2  | 2 |   | 5 | : | 42 | 0 | 0 |  |

Table D.98: CPP model solution of the 10C.t instance.



OBJECTIVE: 47  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 36

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 4 | 5 |   |
| 2  | 0 | 1 | 3 | 5 |
| 3  | 0 | 3 | 5 |   |
| 4  | 0 | 2 | 5 |   |
| 5  | 0 | 3 | 5 |   |
| 6  | 0 | 1 | 5 |   |
| 7  | 0 | 3 | 5 |   |
| 8  | 0 | 1 | 5 |   |
| 9  | 0 | 4 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |    |    |   |   |   |    |    |   |            |
|----|-----------|-----------|-----------|-----------|---|---|---|----|----|---|---|---|----|----|---|------------|
| c  | i         | p         | s         | m         | i | p | s | m  | i  | p | s | m | i  | p  | s | m          |
| 1  | 0         | :         | 1         | 0         | 3 | 4 | : | 4  | 23 | 2 | 5 | : | 29 | 0  | 0 |            |
| 2  | 0         | :         | 5         | 0         | 1 | 1 | : | 6  | 12 | 1 | 3 | : | 19 | 25 | 1 | 5 : 45 0 0 |
| 3  | 0         | :         | 9         | 0         | 2 | 3 | : | 11 | 45 | 1 | 5 | : | 57 | 0  | 0 |            |
| 4  | 0         | :         | 13        | 0         | 2 | 2 | : | 15 | 15 | 3 | 5 | : | 33 | 0  | 0 |            |
| 5  | 0         | :         | 17        | 0         | 2 | 3 | : | 19 | 29 | 1 | 5 | : | 49 | 0  | 0 |            |
| 6  | 0         | :         | 21        | 0         | 1 | 1 | : | 22 | 37 | 2 | 5 | : | 61 | 0  | 0 |            |
| 7  | 0         | :         | 25        | 0         | 2 | 3 | : | 27 | 9  | 1 | 5 | : | 37 | 0  | 0 |            |
| 8  | 0         | :         | 29        | 0         | 1 | 1 | : | 30 | 21 | 2 | 5 | : | 53 | 0  | 0 |            |
| 9  | 0         | :         | 33        | 0         | 3 | 4 | : | 36 | 27 | 2 | 5 | : | 65 | 0  | 0 |            |
| 10 | 0         | :         | 37        | 0         | 1 | 1 | : | 38 | 1  | 2 | 5 | : | 41 | 0  | 0 |            |

Table D.99: CPP model solution of the 10M1.t instance.

OBJECTIVE: 50  
Total number container positionings (sum of xp's) = 12  
Total moving time (sum of xm's) = 38

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 4 | 5 |   |
| 2  | 0 | 3 | 4 | 5 |
| 3  | 0 | 2 | 5 |   |
| 4  | 0 | 1 | 5 |   |
| 5  | 0 | 1 | 3 | 5 |
| 6  | 0 | 4 | 5 |   |
| 7  | 0 | 3 | 5 |   |
| 8  | 0 | 3 | 5 |   |
| 9  | 0 | 1 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |    |    |   |   |   |    |    |   |            |
|----|-----------|-----------|-----------|-----------|---|---|---|----|----|---|---|---|----|----|---|------------|
| c  | i         | p         | s         | m         | i | p | s | m  | i  | p | s | m | i  | p  | s | m          |
| 1  | 0         | :         | 1         | 0         | 3 | 4 | : | 4  | 43 | 2 | 5 | : | 49 | 0  | 0 |            |
| 2  | 0         | :         | 5         | 0         | 2 | 3 | : | 7  | 16 | 1 | 4 | : | 24 | 3  | 2 | 5 : 29 0 0 |
| 3  | 0         | :         | 9         | 0         | 2 | 2 | : | 11 | 39 | 3 | 5 | : | 53 | 0  | 0 |            |
| 4  | 0         | :         | 13        | 0         | 1 | 1 | : | 14 | 17 | 2 | 5 | : | 33 | 0  | 0 |            |
| 5  | 0         | :         | 17        | 0         | 1 | 1 | : | 18 | 8  | 1 | 3 | : | 27 | 29 | 1 | 5 : 57 0 0 |
| 6  | 0         | :         | 21        | 0         | 3 | 4 | : | 24 | 11 | 2 | 5 | : | 37 | 0  | 0 |            |
| 7  | 0         | :         | 25        | 0         | 2 | 3 | : | 27 | 33 | 1 | 5 | : | 61 | 0  | 0 |            |
| 8  | 0         | :         | 29        | 0         | 2 | 3 | : | 31 | 9  | 1 | 5 | : | 41 | 0  | 0 |            |
| 9  | 0         | :         | 33        | 0         | 1 | 1 | : | 34 | 29 | 2 | 5 | : | 65 | 0  | 0 |            |
| 10 | 0         | :         | 37        | 0         | 1 | 1 | : | 38 | 5  | 2 | 5 | : | 45 | 0  | 0 |            |

Table D.100: CPP model solution of the 10M2.t instance.

OBJECTIVE: 45  
 Total number container positionings (sum of xp's) = 11  
 Total moving time (sum of xm's) = 34

Sequence  

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 3 | 5 |   |
| 2  | 0 | 1 | 5 |   |
| 3  | 0 | 3 | 5 |   |
| 4  | 0 | 1 | 5 |   |
| 5  | 0 | 4 | 5 |   |
| 6  | 0 | 1 | 3 | 5 |
| 7  | 0 | 1 | 5 |   |
| 8  | 0 | 2 | 5 |   |
| 9  | 0 | 1 | 5 |   |
| 10 | 0 | 3 | 5 |   |

Positioning, storage, and moving times  

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |    |   |   |   |   |    |    |   |   |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|---|----|----|---|---|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p  | s | m | i | p | s  | m  |   |   |
| 1  | 0         | :         | 1         | 0         | 2 | : | 3 | : | 3  | 33 | 1 | : | 5 | : | 37 | 0  | 0 | : |
| 2  | 0         | :         | 5         | 0         | 1 | : | 1 | : | 6  | 49 | 2 | : | 5 | : | 57 | 0  | 0 | : |
| 3  | 0         | :         | 9         | 0         | 2 | : | 3 | : | 11 | 17 | 1 | : | 5 | : | 29 | 0  | 0 | : |
| 4  | 0         | :         | 13        | 0         | 1 | : | 1 | : | 14 | 33 | 2 | : | 5 | : | 49 | 0  | 0 | : |
| 5  | 0         | :         | 17        | 0         | 3 | : | 4 | : | 20 | 19 | 2 | : | 5 | : | 41 | 0  | 0 | : |
| 6  | 0         | :         | 21        | 0         | 1 | : | 1 | : | 22 | 24 | 1 | : | 3 | : | 47 | 13 | 1 | : |
| 7  | 0         | :         | 25        | 0         | 1 | : | 1 | : | 26 | 5  | 2 | : | 5 | : | 33 | 0  | 0 | : |
| 8  | 0         | :         | 29        | 0         | 2 | : | 2 | : | 31 | 19 | 3 | : | 5 | : | 53 | 0  | 0 | : |
| 9  | 0         | :         | 33        | 0         | 1 | : | 1 | : | 34 | 9  | 2 | : | 5 | : | 45 | 0  | 0 | : |
| 10 | 0         | :         | 37        | 0         | 2 | : | 3 | : | 39 | 25 | 1 | : | 5 | : | 65 | 0  | 0 | : |

Table D.101: CPP model solution of the 10M3.t instance.

OBJECTIVE: 46  
 Total number container positionings (sum of xp's) = 10  
 Total moving time (sum of xm's) = 36

Sequence  

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 4 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 3 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 2 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 2 | 5 |
| 9  | 0 | 3 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times  

| n  | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |   |    |    |   |   |   |   |    |   |   |   |
|----|-----------|-----------|-----------|---|---|---|---|---|----|----|---|---|---|---|----|---|---|---|
| c  | i         | p         | s         | m | i | p | s | m | i  | p  | s | m |   |   |    |   |   |   |
| 1  | 0         | :         | 1         | 0 | 1 | : | 1 | : | 2  | 65 | 2 | : | 5 | : | 69 | 0 | 0 | : |
| 2  | 0         | :         | 5         | 0 | 3 | : | 4 | : | 8  | 35 | 2 | : | 5 | : | 45 | 0 | 0 | : |
| 3  | 0         | :         | 9         | 0 | 1 | : | 1 | : | 10 | 45 | 2 | : | 5 | : | 57 | 0 | 0 | : |
| 4  | 0         | :         | 13        | 0 | 2 | : | 3 | : | 15 | 17 | 1 | : | 5 | : | 33 | 0 | 0 | : |
| 5  | 0         | :         | 17        | 0 | 1 | : | 1 | : | 18 | 29 | 2 | : | 5 | : | 49 | 0 | 0 | : |
| 6  | 0         | :         | 21        | 0 | 2 | : | 2 | : | 23 | 35 | 3 | : | 5 | : | 61 | 0 | 0 | : |
| 7  | 0         | :         | 25        | 0 | 1 | : | 1 | : | 26 | 9  | 2 | : | 5 | : | 37 | 0 | 0 | : |
| 8  | 0         | :         | 29        | 0 | 2 | : | 2 | : | 31 | 19 | 3 | : | 5 | : | 53 | 0 | 0 | : |
| 9  | 0         | :         | 33        | 0 | 2 | : | 3 | : | 35 | 29 | 1 | : | 5 | : | 65 | 0 | 0 | : |
| 10 | 0         | :         | 37        | 0 | 1 | : | 1 | : | 38 | 1  | 2 | : | 5 | : | 41 | 0 | 0 | : |

Table D.102: CPP model solution of the 10M4.t instance.

OBJECTIVE: 44  
 Total number container positionings (sum of xp's) = 12  
 Total moving time (sum of xm's) = 32

Sequence  
 c | 1 2 3 4  
 -----  
 1 | 0 1 5  
 2 | 0 1 5  
 3 | 0 1 5  
 4 | 0 3 5  
 5 | 0 1 5  
 6 | 0 3 5  
 7 | 0 2 5  
 8 | 0 1 3 5  
 9 | 0 1 3 5  
 10 | 0 3 5

Positioning, storage, and moving times  
 n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
 c | i : p s m | i : p s m | i : p s m | i : p s m |  
 -----  
 1 | 0 : 1 0 1 | 1 : 2 57 2 | 5 : 61 0 0 |  
 2 | 0 : 5 0 1 | 1 : 6 45 2 | 5 : 53 0 0 |  
 3 | 0 : 9 0 1 | 1 : 10 33 2 | 5 : 45 0 0 |  
 4 | 0 : 13 0 2 | 3 : 15 21 1 | 5 : 37 0 0 |  
 5 | 0 : 17 0 1 | 1 : 18 9 2 | 5 : 29 0 0 |  
 6 | 0 : 21 0 2 | 3 : 23 9 1 | 5 : 33 0 0 |  
 7 | 0 : 25 0 2 | 2 : 27 11 3 | 5 : 41 0 0 |  
 8 | 0 : 29 0 1 | 1 : 30 9 1 | 3 : 40 8 1 | 5 : 49 0 0 |  
 9 | 0 : 33 0 1 | 1 : 34 4 1 | 3 : 39 17 1 | 5 : 57 0 0 |  
 10 | 0 : 37 0 2 | 3 : 39 25 1 | 5 : 65 0 0 |

Table D.103: CPP model solution of the 10M5.t instance.

OBJECTIVE: 42  
 Total number container positionings (sum of xp's) = 12  
 Total moving time (sum of xm's) = 30

Sequence  
 c | 1 2 3 4  
 -----  
 1 | 0 1 3 5  
 2 | 0 3 5  
 3 | 0 1 5  
 4 | 0 3 5  
 5 | 0 1 3 5  
 6 | 0 1 5  
 7 | 0 3 5  
 8 | 0 3 5  
 9 | 0 1 5  
 10 | 0 1 5

Positioning, storage, and moving times  
 n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
 c | i : p s m | i : p s m | i : p s m | i : p s m |  
 -----  
 1 | 0 : 1 0 1 | 1 : 2 4 1 | 3 : 7 49 1 | 5 : 57 0 0 |  
 2 | 0 : 5 0 2 | 3 : 7 53 1 | 5 : 61 0 0 |  
 3 | 0 : 9 0 1 | 1 : 10 21 2 | 5 : 33 0 0 |  
 4 | 0 : 13 0 2 | 3 : 15 37 1 | 5 : 53 0 0 |  
 5 | 0 : 17 0 1 | 1 : 18 12 1 | 3 : 31 9 1 | 5 : 41 0 0 |  
 6 | 0 : 21 0 1 | 1 : 22 5 2 | 5 : 29 0 0 |  
 7 | 0 : 25 0 2 | 3 : 27 21 1 | 5 : 49 0 0 |  
 8 | 0 : 29 0 2 | 3 : 31 5 1 | 5 : 37 0 0 |  
 9 | 0 : 33 0 1 | 1 : 34 29 2 | 5 : 65 0 0 |  
 10 | 0 : 37 0 1 | 1 : 38 5 2 | 5 : 45 0 0 |

Table D.104: CPP model solution of the 10M6.t instance.

OBJECTIVE: 43  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 32

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 1 | 5 |   |
| 2  | 0 | 3 | 5 |   |
| 3  | 0 | 1 | 3 | 5 |
| 4  | 0 | 3 | 5 |   |
| 5  | 0 | 1 | 5 |   |
| 6  | 0 | 2 | 5 |   |
| 7  | 0 | 3 | 5 |   |
| 8  | 0 | 1 | 5 |   |
| 9  | 0 | 3 | 5 |   |
| 10 | 0 | 3 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |    |   |   |   |   |    |   |   |   |   |   |    |   |   |   |   |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|---|----|---|---|---|---|---|----|---|---|---|---|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p  | s | m | i | p | s  | m |   |   |   |   |    |   |   |   |   |
| 1  | 0         | :         | 1         | 0         | 1 | : | 1 | : | 2  | 49 | 2 | : | 5 | : | 53 | 0 | 0 | : |   |   |    |   |   |   |   |
| 2  | 0         | :         | 5         | 0         | 2 | : | 3 | : | 7  | 57 | 1 | : | 5 | : | 65 | 0 | 0 | : |   |   |    |   |   |   |   |
| 3  | 0         | :         | 9         | 0         | 1 | : | 1 | : | 10 | 40 | 1 | : | 3 | : | 51 | 5 | 1 | : | 5 | : | 57 | 0 | 0 | : |   |
| 4  | 0         | :         | 13        | 0         | 2 | : | 3 | : | 15 | 17 | 1 | : | 5 | : | 33 | 0 | 0 | : |   |   |    |   |   |   | : |
| 5  | 0         | :         | 17        | 0         | 1 | : | 1 | : | 18 | 29 | 2 | : | 5 | : | 49 | 0 | 0 | : |   |   |    |   |   |   | : |
| 6  | 0         | :         | 21        | 0         | 2 | : | 2 | : | 23 | 11 | 3 | : | 5 | : | 37 | 0 | 0 | : |   |   |    |   |   |   | : |
| 7  | 0         | :         | 25        | 0         | 2 | : | 3 | : | 27 | 1  | 1 | : | 5 | : | 29 | 0 | 0 | : |   |   |    |   |   |   | : |
| 8  | 0         | :         | 29        | 0         | 1 | : | 1 | : | 30 | 9  | 2 | : | 5 | : | 41 | 0 | 0 | : |   |   |    |   |   |   | : |
| 9  | 0         | :         | 33        | 0         | 2 | : | 3 | : | 35 | 25 | 1 | : | 5 | : | 61 | 0 | 0 | : |   |   |    |   |   |   | : |
| 10 | 0         | :         | 37        | 0         | 2 | : | 3 | : | 39 | 5  | 1 | : | 5 | : | 45 | 0 | 0 | : |   |   |    |   |   |   | : |

Table D.105: CPP model solution of the 10M7.t instance.

OBJECTIVE: 45  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 34

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 2 | 5 |   |
| 2  | 0 | 3 | 5 |   |
| 3  | 0 | 1 | 5 |   |
| 4  | 0 | 1 | 5 |   |
| 5  | 0 | 1 | 3 | 5 |
| 6  | 0 | 3 | 5 |   |
| 7  | 0 | 4 | 5 |   |
| 8  | 0 | 3 | 5 |   |
| 9  | 0 | 3 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |    |   |   |   |   |    |   |   |   |   |   |    |   |   |   |   |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|---|----|---|---|---|---|---|----|---|---|---|---|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p  | s | m | i | p | s  | m |   |   |   |   |    |   |   |   |   |
| 1  | 0         | :         | 1         | 0         | 2 | : | 2 | : | 3  | 35 | 3 | : | 5 | : | 41 | 0 | 0 | : |   |   |    |   |   |   |   |
| 2  | 0         | :         | 5         | 0         | 2 | : | 3 | : | 7  | 49 | 1 | : | 5 | : | 57 | 0 | 0 | : |   |   |    |   |   |   |   |
| 3  | 0         | :         | 9         | 0         | 1 | : | 1 | : | 10 | 57 | 2 | : | 5 | : | 69 | 0 | 0 | : |   |   |    |   |   |   |   |
| 4  | 0         | :         | 13        | 0         | 1 | : | 1 | : | 14 | 21 | 2 | : | 5 | : | 37 | 0 | 0 | : |   |   |    |   |   |   |   |
| 5  | 0         | :         | 17        | 0         | 1 | : | 1 | : | 18 | 16 | 1 | : | 3 | : | 35 | 9 | 1 | : | 5 | : | 45 | 0 | 0 | : |   |
| 6  | 0         | :         | 21        | 0         | 2 | : | 3 | : | 23 | 29 | 1 | : | 5 | : | 53 | 0 | 0 | : |   |   |    |   |   |   | : |
| 7  | 0         | :         | 25        | 0         | 3 | : | 4 | : | 28 | 31 | 2 | : | 5 | : | 61 | 0 | 0 | : |   |   |    |   |   |   | : |
| 8  | 0         | :         | 29        | 0         | 2 | : | 3 | : | 31 | 1  | 1 | : | 5 | : | 33 | 0 | 0 | : |   |   |    |   |   |   | : |
| 9  | 0         | :         | 33        | 0         | 2 | : | 3 | : | 35 | 13 | 1 | : | 5 | : | 49 | 0 | 0 | : |   |   |    |   |   |   | : |
| 10 | 0         | :         | 37        | 0         | 1 | : | 1 | : | 38 | 25 | 2 | : | 5 | : | 65 | 0 | 0 | : |   |   |    |   |   |   | : |

Table D.106: CPP model solution of the 10M8.t instance.

OBJECTIVE: 45  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 34

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 2 | 5 |   |
| 2  | 0 | 1 | 5 |   |
| 3  | 0 | 3 | 5 |   |
| 4  | 0 | 2 | 5 |   |
| 5  | 0 | 1 | 5 |   |
| 6  | 0 | 1 | 5 |   |
| 7  | 0 | 3 | 5 |   |
| 8  | 0 | 1 | 3 | 5 |
| 9  | 0 | 3 | 5 |   |
| 10 | 0 | 3 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |    |   |   |   |   |    |    |   |  |   |   |    |   |   |  |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|---|----|----|---|--|---|---|----|---|---|--|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p  | s | m | i | p | s  | m  |   |  |   |   |    |   |   |  |
| 1  | 0         | :         | 1         | 0         | 2 |   | 2 | : | 3  | 43 | 3 |   | 5 | : | 49 | 0  | 0 |  |   |   |    |   |   |  |
| 2  | 0         | :         | 5         | 0         | 1 |   | 1 | : | 6  | 57 | 2 |   | 5 | : | 65 | 0  | 0 |  |   |   |    |   |   |  |
| 3  | 0         | :         | 9         | 0         | 2 |   | 3 | : | 11 | 61 | 1 |   | 5 | : | 73 | 0  | 0 |  |   |   |    |   |   |  |
| 4  | 0         | :         | 13        | 0         | 2 |   | 2 | : | 15 | 23 | 3 |   | 5 | : | 41 | 0  | 0 |  |   |   |    |   |   |  |
| 5  | 0         | :         | 17        | 0         | 1 |   | 1 | : | 18 | 41 | 2 |   | 5 | : | 61 | 0  | 0 |  |   |   |    |   |   |  |
| 6  | 0         | :         | 21        | 0         | 1 |   | 1 | : | 22 | 21 | 2 |   | 5 | : | 45 | 0  | 0 |  |   |   |    |   |   |  |
| 7  | 0         | :         | 25        | 0         | 2 |   | 3 | : | 27 | 41 | 1 |   | 5 | : | 69 | 0  | 0 |  |   |   |    |   |   |  |
| 8  | 0         | :         | 29        | 0         | 1 |   | 1 | : | 30 | 8  | 1 |   | 3 | : | 39 | 13 | 1 |  | 5 | : | 53 | 0 | 0 |  |
| 9  | 0         | :         | 33        | 0         | 2 |   | 3 | : | 35 | 1  | 1 |   | 5 | : | 37 | 0  | 0 |  |   |   |    |   |   |  |
| 10 | 0         | :         | 37        | 0         | 2 |   | 3 | : | 39 | 17 | 1 |   | 5 | : | 57 | 0  | 0 |  |   |   |    |   |   |  |

Table D.107: CPP model solution of the 10M9.t instance.

OBJECTIVE: 45  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 34

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 1 | 5 |   |
| 2  | 0 | 1 | 3 | 5 |
| 3  | 0 | 3 | 5 |   |
| 4  | 0 | 3 | 5 |   |
| 5  | 0 | 1 | 5 |   |
| 6  | 0 | 3 | 5 |   |
| 7  | 0 | 2 | 5 |   |
| 8  | 0 | 2 | 5 |   |
| 9  | 0 | 1 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |    |   |   |   |   |    |    |   |  |   |   |    |   |   |  |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|----|---|---|---|---|----|----|---|--|---|---|----|---|---|--|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p  | s | m | i | p | s  | m  |   |  |   |   |    |   |   |  |
| 1  | 0         | :         | 1         | 0         | 1 |   | 1 | : | 2  | 66 | 2 |   | 5 | : | 70 | 0  | 0 |  |   |   |    |   |   |  |
| 2  | 0         | :         | 5         | 0         | 1 |   | 1 | : | 6  | 8  | 1 |   | 3 | : | 15 | 34 | 1 |  | 5 | : | 50 | 0 | 0 |  |
| 3  | 0         | :         | 9         | 0         | 2 |   | 3 | : | 11 | 62 | 1 |   | 5 | : | 74 | 0  | 0 |  |   |   |    |   |   |  |
| 4  | 0         | :         | 13        | 0         | 2 |   | 3 | : | 15 | 38 | 1 |   | 5 | : | 54 | 0  | 0 |  |   |   |    |   |   |  |
| 5  | 0         | :         | 17        | 0         | 1 |   | 1 | : | 18 | 46 | 2 |   | 5 | : | 66 | 0  | 0 |  |   |   |    |   |   |  |
| 6  | 0         | :         | 21        | 0         | 2 |   | 3 | : | 23 | 22 | 1 |   | 5 | : | 46 | 0  | 0 |  |   |   |    |   |   |  |
| 7  | 0         | :         | 25        | 0         | 2 |   | 2 | : | 27 | 48 | 3 |   | 5 | : | 78 | 0  | 0 |  |   |   |    |   |   |  |
| 8  | 0         | :         | 29        | 0         | 2 |   | 2 | : | 31 | 24 | 3 |   | 5 | : | 58 | 0  | 0 |  |   |   |    |   |   |  |
| 9  | 0         | :         | 33        | 0         | 1 |   | 1 | : | 34 | 26 | 2 |   | 5 | : | 62 | 0  | 0 |  |   |   |    |   |   |  |
| 10 | 0         | :         | 37        | 0         | 1 |   | 1 | : | 38 | 2  | 2 |   | 5 | : | 42 | 0  | 0 |  |   |   |    |   |   |  |

Table D.108: CPP model solution of the 10M10.t instance.

OBJECTIVE: 60  
 Total number container positionings (sum of xp's) = 18  
 Total moving time (sum of xm's) = 42

Sequence  

| c  | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|---|
| 1  | 0 | 2 | 5 |   |   |
| 2  | 0 | 1 | 2 | 3 | 5 |
| 3  | 0 | 1 | 4 | 5 |   |
| 4  | 0 | 1 | 3 | 5 |   |
| 5  | 0 | 1 | 3 | 4 | 5 |
| 6  | 0 | 3 | 4 | 5 |   |
| 7  | 0 | 1 | 3 | 5 |   |
| 8  | 0 | 4 | 5 |   |   |
| 9  | 0 | 3 | 5 |   |   |
| 10 | 0 | 1 | 5 |   |   |

Positioning, storage, and moving times  

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 | Number: 5 |   |   |    |     |   |   |   |     |     |   |   |   |     |     |   |   |   |     |   |   |
|----|-----------|-----------|-----------|-----------|-----------|---|---|----|-----|---|---|---|-----|-----|---|---|---|-----|-----|---|---|---|-----|---|---|
| c  | i         | p         | s         | m         | i         | p | s | m  | i   | p | s | m | i   | p   | s | m |   |     |     |   |   |   |     |   |   |
| 1  | 0         | :         | 1         | 0         | 2         | 2 | : | 3  | 81  | 3 | 5 | : | 87  | 0   | 0 |   |   |     |     |   |   |   |     |   |   |
| 2  | 0         | :         | 5         | 0         | 1         | 1 | : | 6  | 0   | 1 | 2 | : | 7   | 27  | 2 | 3 | : | 36  | 62  | 1 | 5 | : | 99  | 0 | 0 |
| 3  | 0         | :         | 9         | 0         | 1         | 1 | : | 10 | 26  | 2 | 4 | : | 38  | 71  | 2 | 5 | : | 111 | 0   | 0 |   |   |     |   |   |
| 4  | 0         | :         | 13        | 0         | 1         | 1 | : | 14 | 21  | 1 | 3 | : | 36  | 86  | 1 | 5 | : | 123 | 0   | 0 |   |   |     |   |   |
| 5  | 0         | :         | 17        | 0         | 1         | 1 | : | 18 | 4   | 1 | 3 | : | 23  | 8   | 1 | 4 | : | 32  | 101 | 2 | 5 | : | 135 | 0 | 0 |
| 6  | 0         | :         | 21        | 0         | 2         | 1 | : | 23 | 8   | 1 | 4 | : | 32  | 113 | 2 | 5 | : | 147 | 0   | 0 |   |   |     |   |   |
| 7  | 0         | :         | 25        | 0         | 1         | 1 | : | 26 | 8   | 1 | 3 | : | 35  | 123 | 1 | 5 | : | 159 | 0   | 0 |   |   |     |   |   |
| 8  | 0         | :         | 29        | 0         | 3         | 1 | : | 32 | 137 | 2 | 5 | : | 171 | 0   | 0 |   |   |     |     |   |   |   |     |   |   |
| 9  | 0         | :         | 33        | 0         | 2         | 1 | : | 35 | 147 | 1 | 5 | : | 183 | 0   | 0 |   |   |     |     |   |   |   |     |   |   |
| 10 | 0         | :         | 37        | 0         | 1         | 1 | : | 38 | 155 | 2 | 5 | : | 195 | 0   | 0 |   |   |     |     |   |   |   |     |   |   |

Table D.109: CPP model solution of the 10S.s instance.

OBJECTIVE: 40  
 Total number container positionings (sum of xp's) = 10  
 Total moving time (sum of xm's) = 30

Sequence  

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times  

| n  | Number: 1 | Number: 2 | Number: 3 |   |   |   |   |    |     |   |   |   |     |   |   |
|----|-----------|-----------|-----------|---|---|---|---|----|-----|---|---|---|-----|---|---|
| c  | i         | p         | s         | m | i | p | s | m  | i   | p | s | m |     |   |   |
| 1  | 0         | :         | 1         | 0 | 1 | 1 | : | 2  | 230 | 2 | 5 | : | 234 | 0 | 0 |
| 2  | 0         | :         | 5         | 0 | 1 | 1 | : | 6  | 214 | 2 | 5 | : | 222 | 0 | 0 |
| 3  | 0         | :         | 9         | 0 | 1 | 1 | : | 10 | 198 | 2 | 5 | : | 210 | 0 | 0 |
| 4  | 0         | :         | 13        | 0 | 1 | 1 | : | 14 | 182 | 2 | 5 | : | 198 | 0 | 0 |
| 5  | 0         | :         | 17        | 0 | 1 | 1 | : | 18 | 166 | 2 | 5 | : | 186 | 0 | 0 |
| 6  | 0         | :         | 21        | 0 | 1 | 1 | : | 22 | 150 | 2 | 5 | : | 174 | 0 | 0 |
| 7  | 0         | :         | 25        | 0 | 1 | 1 | : | 26 | 134 | 2 | 5 | : | 162 | 0 | 0 |
| 8  | 0         | :         | 29        | 0 | 1 | 1 | : | 30 | 118 | 2 | 5 | : | 150 | 0 | 0 |
| 9  | 0         | :         | 33        | 0 | 1 | 1 | : | 34 | 102 | 2 | 5 | : | 138 | 0 | 0 |
| 10 | 0         | :         | 37        | 0 | 1 | 1 | : | 38 | 86  | 2 | 5 | : | 126 | 0 | 0 |

Table D.110: CPP model solution of the 10C.s instance.

OBJECTIVE: 51  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 40

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 2 | 4 | 5 |
| 2  | 0 | 3 | 5 |   |
| 3  | 0 | 4 | 5 |   |
| 4  | 0 | 3 | 5 |   |
| 5  | 0 | 4 | 5 |   |
| 6  | 0 | 1 | 5 |   |
| 7  | 0 | 4 | 5 |   |
| 8  | 0 | 1 | 5 |   |
| 9  | 0 | 2 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |     |   |   |   |   |     |    |   |  |   |   |    |   |   |  |  |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|-----|---|---|---|---|-----|----|---|--|---|---|----|---|---|--|--|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p   | s | m | i | p | s   | m  |   |  |   |   |    |   |   |  |  |
| 1  | 0         | :         | 1         | 0         | 2 |   | 2 | : | 3  | 24  | 1 |   | 4 | : | 28  | 57 | 2 |  | 5 | : | 87 | 0 | 0 |  |  |
| 2  | 0         | :         | 5         | 0         | 2 |   | 3 | : | 7  | 127 | 1 |   | 5 | : | 135 | 0  | 0 |  |   |   |    |   |   |  |  |
| 3  | 0         | :         | 9         | 0         | 3 |   | 4 | : | 12 | 157 | 2 |   | 5 | : | 171 | 0  | 0 |  |   |   |    |   |   |  |  |
| 4  | 0         | :         | 13        | 0         | 2 |   | 3 | : | 15 | 83  | 1 |   | 5 | : | 99  | 0  | 0 |  |   |   |    |   |   |  |  |
| 5  | 0         | :         | 17        | 0         | 3 |   | 4 | : | 20 | 125 | 2 |   | 5 | : | 147 | 0  | 0 |  |   |   |    |   |   |  |  |
| 6  | 0         | :         | 21        | 0         | 1 |   | 1 | : | 22 | 159 | 2 |   | 5 | : | 183 | 0  | 0 |  |   |   |    |   |   |  |  |
| 7  | 0         | :         | 25        | 0         | 3 |   | 4 | : | 28 | 81  | 2 |   | 5 | : | 111 | 0  | 0 |  |   |   |    |   |   |  |  |
| 8  | 0         | :         | 29        | 0         | 1 |   | 1 | : | 30 | 127 | 2 |   | 5 | : | 159 | 0  | 0 |  |   |   |    |   |   |  |  |
| 9  | 0         | :         | 33        | 0         | 2 |   | 2 | : | 35 | 157 | 3 |   | 5 | : | 195 | 0  | 0 |  |   |   |    |   |   |  |  |
| 10 | 0         | :         | 37        | 0         | 1 |   | 1 | : | 38 | 83  | 2 |   | 5 | : | 123 | 0  | 0 |  |   |   |    |   |   |  |  |

Table D.111: CPP model solution of the 10M1.s instance.

OBJECTIVE: 54  
Total number container positionings (sum of xp's) = 12  
Total moving time (sum of xm's) = 42

Sequence

| c  | 1 | 2 | 3 | 4 |
|----|---|---|---|---|
| 1  | 0 | 2 | 5 |   |
| 2  | 0 | 1 | 3 | 5 |
| 3  | 0 | 3 | 5 |   |
| 4  | 0 | 3 | 5 |   |
| 5  | 0 | 4 | 5 |   |
| 6  | 0 | 2 | 5 |   |
| 7  | 0 | 4 | 2 | 5 |
| 8  | 0 | 4 | 5 |   |
| 9  | 0 | 1 | 5 |   |
| 10 | 0 | 1 | 5 |   |

Positioning, storage, and moving times

| n  | Number: 1 | Number: 2 | Number: 3 | Number: 4 |   |   |   |   |    |     |   |   |   |   |     |    |   |  |   |   |     |   |   |  |  |  |
|----|-----------|-----------|-----------|-----------|---|---|---|---|----|-----|---|---|---|---|-----|----|---|--|---|---|-----|---|---|--|--|--|
| c  | i         | p         | s         | m         | i | p | s | m | i  | p   | s | m | i | p | s   | m  |   |  |   |   |     |   |   |  |  |  |
| 1  | 0         | :         | 1         | 0         | 2 |   | 2 | : | 3  | 141 | 3 |   | 5 | : | 147 | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 2  | 0         | :         | 5         | 0         | 1 |   | 1 | : | 6  | 8   | 1 |   | 3 | : | 15  | 71 | 1 |  | 5 | : | 87  | 0 | 0 |  |  |  |
| 3  | 0         | :         | 9         | 0         | 2 |   | 3 | : | 11 | 147 | 1 |   | 5 | : | 159 | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 4  | 0         | :         | 13        | 0         | 2 |   | 3 | : | 15 | 83  | 1 |   | 5 | : | 99  | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 5  | 0         | :         | 17        | 0         | 3 |   | 4 | : | 20 | 149 | 2 |   | 5 | : | 171 | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 6  | 0         | :         | 21        | 0         | 2 |   | 2 | : | 23 | 85  | 3 |   | 5 | : | 111 | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 7  | 0         | :         | 25        | 0         | 3 |   | 4 | : | 28 | 115 | 1 |   | 2 | : | 144 | 36 | 3 |  | 5 | : | 183 | 0 | 0 |  |  |  |
| 8  | 0         | :         | 29        | 0         | 3 |   | 4 | : | 32 | 89  | 2 |   | 5 | : | 123 | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 9  | 0         | :         | 33        | 0         | 1 |   | 1 | : | 34 | 159 | 2 |   | 5 | : | 195 | 0  | 0 |  |   |   |     |   |   |  |  |  |
| 10 | 0         | :         | 37        | 0         | 1 |   | 1 | : | 38 | 95  | 2 |   | 5 | : | 135 | 0  | 0 |  |   |   |     |   |   |  |  |  |

Table D.112: CPP model solution of the 10M2.s instance.

OBJECTIVE: 46  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 36

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 2 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 2 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 3 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 3 | 5 |
| 9  | 0 | 3 | 5 |
| 10 | 0 | 4 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |   |    |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |   |     |   |   |  |
|----|-----------|---|---|----|---|-----------|---|---|---|----|-----------|---|---|---|---|-----|---|---|--|
| c  | i         | p | s | m  |   | i         | p | s | m |    | i         | p | s | m |   |     |   |   |  |
| 1  |           | 0 | : | 1  | 0 | 2         |   | 2 | : | 3  | 105       | 3 |   | 5 | : | 111 | 0 | 0 |  |
| 2  |           | 0 | : | 5  | 0 | 1         |   | 1 | : | 6  | 163       | 2 |   | 5 | : | 171 | 0 | 0 |  |
| 3  |           | 0 | : | 9  | 0 | 2         |   | 2 | : | 11 | 73        | 3 |   | 5 | : | 87  | 0 | 0 |  |
| 4  |           | 0 | : | 13 | 0 | 1         |   | 1 | : | 14 | 131       | 2 |   | 5 | : | 147 | 0 | 0 |  |
| 5  |           | 0 | : | 17 | 0 | 1         |   | 1 | : | 18 | 103       | 2 |   | 5 | : | 123 | 0 | 0 |  |
| 6  |           | 0 | : | 21 | 0 | 2         |   | 3 | : | 23 | 159       | 1 |   | 5 | : | 183 | 0 | 0 |  |
| 7  |           | 0 | : | 25 | 0 | 1         |   | 1 | : | 26 | 71        | 2 |   | 5 | : | 99  | 0 | 0 |  |
| 8  |           | 0 | : | 29 | 0 | 2         |   | 3 | : | 31 | 127       | 1 |   | 5 | : | 159 | 0 | 0 |  |
| 9  |           | 0 | : | 33 | 0 | 2         |   | 3 | : | 35 | 99        | 1 |   | 5 | : | 135 | 0 | 0 |  |
| 10 |           | 0 | : | 37 | 0 | 3         |   | 4 | : | 40 | 153       | 2 |   | 5 | : | 195 | 0 | 0 |  |

Table D.113: CPP model solution of the 10M3.s instance.

OBJECTIVE: 46  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 36

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 2 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 2 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 3 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 3 | 5 |
| 9  | 0 | 4 | 5 |
| 10 | 0 | 3 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   |   | Number: 2 |   |    |     |   | Number: 3 |   |     |   |   |  |
|----|-----------|---|----|---|---|-----------|---|----|-----|---|-----------|---|-----|---|---|--|
| c  | i         | p | s  | m |   | i         | p | s  | m   |   | i         | p | s   | m |   |  |
| 1  | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 203 | 2 | 5         | : | 207 | 0 | 0 |  |
| 2  | 0         | : | 5  | 0 | 2 | 2         | : | 7  | 125 | 3 | 5         | : | 135 | 0 | 0 |  |
| 3  | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 159 | 2 | 5         | : | 171 | 0 | 0 |  |
| 4  | 0         | : | 13 | 0 | 2 | 2         | : | 15 | 81  | 3 | 5         | : | 99  | 0 | 0 |  |
| 5  | 0         | : | 17 | 0 | 1 | 1         | : | 18 | 127 | 2 | 5         | : | 147 | 0 | 0 |  |
| 6  | 0         | : | 21 | 0 | 2 | 3         | : | 23 | 159 | 1 | 5         | : | 183 | 0 | 0 |  |
| 7  | 0         | : | 25 | 0 | 1 | 1         | : | 26 | 83  | 2 | 5         | : | 111 | 0 | 0 |  |
| 8  | 0         | : | 29 | 0 | 2 | 3         | : | 31 | 127 | 1 | 5         | : | 159 | 0 | 0 |  |
| 9  | 0         | : | 33 | 0 | 3 | 4         | : | 36 | 157 | 2 | 5         | : | 195 | 0 | 0 |  |
| 10 | 0         | : | 37 | 0 | 2 | 3         | : | 39 | 83  | 1 | 5         | : | 123 | 0 | 0 |  |

Table D.114: CPP model solution of the 10M4.s instance.



OBJECTIVE: 46  
 Total number container positionings (sum of xp's) = 12  
 Total moving time (sum of xm's) = 34

Sequence  
 c | 1 2 3 4  
 -----  
 1 | 0 1 5  
 2 | 0 1 5  
 3 | 0 1 5  
 4 | 0 1 5  
 5 | 0 1 5  
 6 | 0 1 3 5  
 7 | 0 1 3 5  
 8 | 0 2 5  
 9 | 0 4 5  
 10 | 0 3 5

Positioning, storage, and moving times  
 n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
 c | i : p s m | i : p s m | i : p s m | i : p s m |  
 -----  
 1 | 0 : 1 0 1 | 1 : 2 179 2 | 5 : 183 0 0 |  
 2 | 0 : 5 0 1 | 1 : 6 151 2 | 5 : 159 0 0 |  
 3 | 0 : 9 0 1 | 1 : 10 123 2 | 5 : 135 0 0 |  
 4 | 0 : 13 0 1 | 1 : 14 95 2 | 5 : 111 0 0 |  
 5 | 0 : 17 0 1 | 1 : 18 67 2 | 5 : 87 0 0 |  
 6 | 0 : 21 0 1 | 1 : 22 17 1 | 3 : 40 58 1 | 5 : 99 0 0 |  
 7 | 0 : 25 0 1 | 1 : 26 12 1 | 3 : 39 83 1 | 5 : 123 0 0 |  
 8 | 0 : 29 0 2 | 2 : 31 113 3 | 5 : 147 0 0 |  
 9 | 0 : 33 0 3 | 4 : 36 133 2 | 5 : 171 0 0 |  
 10 | 0 : 37 0 2 | 3 : 39 155 1 | 5 : 195 0 0 |

Table D.115: CPP model solution of the 10M5.s instance.

OBJECTIVE: 45  
 Total number container positionings (sum of xp's) = 11  
 Total moving time (sum of xm's) = 34

Sequence  
 c | 1 2 3 4  
 -----  
 1 | 0 1 5  
 2 | 0 3 5  
 3 | 0 2 5  
 4 | 0 1 5  
 5 | 0 1 5  
 6 | 0 1 5  
 7 | 0 3 5  
 8 | 0 1 3 5  
 9 | 0 4 5  
 10 | 0 3 5

Positioning, storage, and moving times  
 n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
 c | i : p s m | i : p s m | i : p s m | i : p s m |  
 -----  
 1 | 0 : 1 0 1 | 1 : 2 167 2 | 5 : 171 0 0 |  
 2 | 0 : 5 0 2 | 3 : 7 175 1 | 5 : 183 0 0 |  
 3 | 0 : 9 0 2 | 2 : 11 85 3 | 5 : 99 0 0 |  
 4 | 0 : 13 0 1 | 1 : 14 143 2 | 5 : 159 0 0 |  
 5 | 0 : 17 0 1 | 1 : 18 103 2 | 5 : 123 0 0 |  
 6 | 0 : 21 0 1 | 1 : 22 63 2 | 5 : 87 0 0 |  
 7 | 0 : 25 0 2 | 3 : 27 119 1 | 5 : 147 0 0 |  
 8 | 0 : 29 0 1 | 1 : 30 8 1 | 3 : 39 71 1 | 5 : 111 0 0 |  
 9 | 0 : 33 0 3 | 4 : 36 157 2 | 5 : 195 0 0 |  
 10 | 0 : 37 0 2 | 3 : 39 95 1 | 5 : 135 0 0 |

Table D.116: CPP model solution of the 10M6.s instance.

OBJECTIVE: 45  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 34

Sequence  
c | 1 2 3 4  
-----  
1 | 0 1 5  
2 | 0 3 5  
3 | 0 3 5  
4 | 0 1 3 5  
5 | 0 3 5  
6 | 0 3 5  
7 | 0 3 5  
8 | 0 1 5  
9 | 0 2 5  
10 | 0 2 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
c | i : p s m | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 1 : 2 155 2 | 5 : 159 0 0 |  
2 | 0 : 5 0 2 | 3 : 7 187 1 | 5 : 195 0 0 |  
3 | 0 : 9 0 2 | 3 : 11 159 1 | 5 : 171 0 0 |  
4 | 0 : 13 0 1 | 1 : 14 8 1 | 3 : 23 75 1 | 5 : 99 0 0 |  
5 | 0 : 17 0 2 | 3 : 19 127 1 | 5 : 147 0 0 |  
6 | 0 : 21 0 2 | 3 : 23 87 1 | 5 : 111 0 0 |  
7 | 0 : 25 0 2 | 3 : 27 59 1 | 5 : 87 0 0 |  
8 | 0 : 29 0 1 | 1 : 30 91 2 | 5 : 123 0 0 |  
9 | 0 : 33 0 2 | 2 : 35 145 3 | 5 : 183 0 0 |  
10 | 0 : 37 0 2 | 2 : 39 93 3 | 5 : 135 0 0 |

Table D.117: CPP model solution of the 10M7.s instance.

OBJECTIVE: 49  
Total number container positionings (sum of xp's) = 11  
Total moving time (sum of xm's) = 38

Sequence  
c | 1 2 3 4  
-----  
1 | 0 3 5  
2 | 0 4 5  
3 | 0 1 5  
4 | 0 3 5  
5 | 0 4 5  
6 | 0 1 5  
7 | 0 2 5  
8 | 0 3 5  
9 | 0 1 5  
10 | 0 2 1 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
c | i : p s m | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 3 : 3 119 1 | 5 : 123 0 0 |  
2 | 0 : 5 0 3 | 4 : 8 161 2 | 5 : 171 0 0 |  
3 | 0 : 9 0 1 | 1 : 10 195 2 | 5 : 207 0 0 |  
4 | 0 : 13 0 2 | 3 : 15 95 1 | 5 : 111 0 0 |  
5 | 0 : 17 0 3 | 4 : 20 113 2 | 5 : 135 0 0 |  
6 | 0 : 21 0 1 | 1 : 22 135 2 | 5 : 159 0 0 |  
7 | 0 : 25 0 2 | 2 : 27 153 3 | 5 : 183 0 0 |  
8 | 0 : 29 0 2 | 3 : 31 67 1 | 5 : 99 0 0 |  
9 | 0 : 33 0 1 | 1 : 34 111 2 | 5 : 147 0 0 |  
10 | 0 : 37 0 2 | 2 : 39 117 1 | 1 : 157 36 2 | 5 : 195 0 0 |

Table D.118: CPP model solution of the 10M8.s instance.

OBJECTIVE: 44  
Total number container positionings (sum of xp's) = 12  
Total moving time (sum of xm's) = 32

Sequence  
c | 1 2 3 4  
-----  
1 | 0 2 5  
2 | 0 3 5  
3 | 0 1 5  
4 | 0 1 3 5  
5 | 0 3 5  
6 | 0 3 5  
7 | 0 1 5  
8 | 0 1 5  
9 | 0 3 5  
10 | 0 1 3 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
c | i : p s m | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 2 | 2 : 3 141 3 | 5 : 147 0 0 |  
2 | 0 : 5 0 2 | 3 : 7 187 1 | 5 : 195 0 0 |  
3 | 0 : 9 0 1 | 1 : 10 207 2 | 5 : 219 0 0 |  
4 | 0 : 13 0 1 | 1 : 14 8 1 | 3 : 23 99 1 | 5 : 123 0 0 |  
5 | 0 : 17 0 2 | 3 : 19 163 1 | 5 : 183 0 0 |  
6 | 0 : 21 0 2 | 3 : 23 111 1 | 5 : 135 0 0 |  
7 | 0 : 25 0 1 | 1 : 26 179 2 | 5 : 207 0 0 |  
8 | 0 : 29 0 1 | 1 : 30 127 2 | 5 : 159 0 0 |  
9 | 0 : 33 0 2 | 3 : 35 75 1 | 5 : 111 0 0 |  
10 | 0 : 37 0 1 | 1 : 38 118 1 | 3 : 157 13 1 | 5 : 171 0 0 |

Table D.119: CPP model solution of the 10M9.s instance.

OBJECTIVE: 44  
Total number container positionings (sum of xp's) = 12  
Total moving time (sum of xm's) = 32

Sequence  
c | 1 2 3 4  
-----  
1 | 0 1 5  
2 | 0 1 5  
3 | 0 3 5  
4 | 0 1 3 5  
5 | 0 3 5  
6 | 0 1 5  
7 | 0 2 5  
8 | 0 1 3 5  
9 | 0 3 5  
10 | 0 1 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |  
c | i : p s m | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 1 : 2 206 2 | 5 : 210 0 0 |  
2 | 0 : 5 0 1 | 1 : 6 142 2 | 5 : 150 0 0 |  
3 | 0 : 9 0 2 | 3 : 11 210 1 | 5 : 222 0 0 |  
4 | 0 : 13 0 1 | 1 : 14 123 1 | 3 : 138 23 1 | 5 : 162 0 0 |  
5 | 0 : 17 0 2 | 3 : 19 178 1 | 5 : 198 0 0 |  
6 | 0 : 21 0 1 | 1 : 22 114 2 | 5 : 138 0 0 |  
7 | 0 : 25 0 2 | 2 : 27 204 3 | 5 : 234 0 0 |  
8 | 0 : 29 0 1 | 1 : 30 4 1 | 3 : 35 138 1 | 5 : 174 0 0 |  
9 | 0 : 33 0 2 | 3 : 35 150 1 | 5 : 186 0 0 |  
10 | 0 : 37 0 1 | 1 : 38 86 2 | 5 : 126 0 0 |

Table D.120: CPP model solution of the 10M10.s instance.

# Appendix E

## CPPT model solutions

### E.1 Small-scale class instances

```
OBJECTIVE: 14
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 10

0 m 1 1 1 1 m 3 - - - -
- 0 m 1 1 m 2 m m 3 - -
- - 0 m m 2 2 2 2 m m 3
```

Table E.1: CPPT model solution of the 3S.t instance.

```
OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 m 3
- 0 m 1 1 1 1 1 m 3 -
- - 0 m m 2 m m 3 - -
```

Table E.2: CPPT model solution of the 3C.t instance.

```
OBJECTIVE: 15
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 10

0 m 1 1 1 1 m 3 - - - -
- 0 m 1 m 2 2 2 2 m m 3
- - 0 m 1 m 2 2 m m 3 - -
```

Table E.3: CPPT model solution of the 3M1A.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m m 2 2 2 2 m m 3 - - -
- - 0 m 1 1 1 1 1 1 m 3
- - - - 0 m 1 1 1 1 m 3 -

```

Table E.4: CPPT model solution of the 3M1B.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m 1 1 1 1 m 3 - -
- 0 m 1 1 m 3 - - -
- - 0 m m 2 2 m m 3

```

Table E.5: CPPT model solution of the 3M2A.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 m 3
- 0 m 1 1 m 3 - - -
- - 0 m m 2 m m 3 -

```

Table E.6: CPPT model solution of the 3M2B.t instance.

```

OBJECTIVE: 12
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 1 m 3 - -
- 0 m 1 m 2 2 2 2 2 m m 3
- - 0 m 1 1 1 m 3 - - - -

```

Table E.7: CPPT model solution of the 3M3A.t instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m m 2 2 2 2 2 m m 3 -
- - 0 m 1 1 1 1 1 m 3
- - - 0 m 1 m 3 - - -

```

Table E.8: CPPT model solution of the 3M3B.t instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 10

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- 0 m 1 1 1 1 m 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - - - -
- - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3

```

Table E.9: CPPT model solution of the 3S.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3
- 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - -
- - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - - - -

```

Table E.10: CPPT model solution of the 3C.s instance.

```

OBJECTIVE: 15
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 10

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- 0 m 1 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3
- - 0 m 1 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - - - -

```

Table E.11: CPPT model solution of the 3M1A.s instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - - - -
- - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - -

```

Table E.12: CPPT model solution of the **3M1B.s** instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3

```

Table E.13: CPPT model solution of the **3M2A.s** instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3
- 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - -

```

Table E.14: CPPT model solution of the **3M2B.s** instance.

```

OBJECTIVE: 12
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 8

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- 0 m 1 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3
- - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -

```

Table E.15: CPPT model solution of the **3M3A.s** instance.

```

OBJECTIVE: 11
Total number container positionings (sum of xm's) = 3
Total moving time (sum of y's) = 8

0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - -
- - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - - - - -

```

Table E.16: CPPT model solution of the 3M3B.s instance.

```

OBJECTIVE: 20
Total number container positionings (sum of xm's) = 6
Total moving time (sum of y's) = 14

0 m m 2 2 2 m 1 1 1 1 m 3 - - - - -
- - 0 m 1 1 1 1 1 1 1 1 1 m 3 - - - -
- - - - 0 m 1 1 1 1 1 1 m 2 m m 3 - -
- - - - - 0 m m 2 2 2 2 2 2 m m 3

```

Table E.17: CPPT model solution of the 4S.t instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 10

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3
- - 0 m 1 1 1 1 1 1 1 1 1 1 m 3 - -
- - - - 0 m m 2 2 2 2 2 m m 3 - - - -
- - - - - 0 m 1 1 1 m 3 - - - - -

```

Table E.18: CPPT model solution of the 4C.t instance.

```

OBJECTIVE: 16
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 12

0 m m 2 2 2 2 2 2 m m 3 - - - - -
- - 0 m 1 1 1 1 1 1 1 1 1 m 3 - -
- - - - 0 m 1 1 1 1 1 1 m 3 - - - -
- - - - - 0 m m 2 2 2 2 2 m m 3

```

Table E.19: CPPT model solution of the 4M1A.t instance.



```

OBJECTIVE: 17
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 12

0 m m 2 2 2 2 2 m m 3 - - - - -
- - 0 m 1 1 1 1 1 1 1 1 m 3 - -
- - - - 0 m 1 1 1 1 1 1 m 2 m m 3
- - - - - 0 m 1 1 1 m 3 - - - -

```

Table E.20: CPPT model solution of the **4M1B.t** instance.

```

OBJECTIVE: 17
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 12

0 m 1 1 1 1 1 1 1 1 m 3 - - - -
- - 0 m 1 1 1 1 1 m 3 - - - - -
- - - - 0 m m 2 2 2 2 2 m m 3 - -
- - - - - 0 m m 2 m 1 1 1 1 m 3

```

Table E.21: CPPT model solution of the **4M2A.t** instance.

```

OBJECTIVE: 16
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 12

0 m 1 1 1 1 1 1 1 1 1 m 3 - -
- - 0 m 1 1 1 1 1 m 3 - - - - -
- - - - 0 m m 2 2 2 2 2 2 m m 3
- - - - - 0 m m 2 m m 3 - - - -

```

Table E.22: CPPT model solution of the **4M2B.t** instance.

```

OBJECTIVE: 16
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 12

0 m 1 1 1 1 1 1 1 1 m 3 - - - -
- - 0 m m 2 2 2 2 2 2 2 2 m m 3
- - - - 0 m 1 1 1 m 3 - - - - -
- - - - - 0 m m 2 2 2 m m 3 - -

```

Table E.23: CPPT model solution of the **4M3A.t** instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 10

0 m 1 1 1 1 1 1 1 1 1 1 1 m 3
-- 0 m m 2 2 2 2 m m 3 - - - -
- - - - 0 m 1 1 m 3 - - - - -
- - - - - 0 m 1 1 1 1 m 3 - -

```

Table E.24: CPPT model solution of the **4M3B.t** instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 10

0 m m 2 2 2 2 2 2 2 m m 3 - - - -
-- 0 m 1 1 1 1 1 1 1 1 1 1 m 3
- - - - 0 m 1 1 1 1 1 1 m 3 - -
- - - - - 0 m 1 m 3 - - - - -

```

Table E.25: CPPT model solution of the **4M4A.t** instance.

```

OBJECTIVE: 14
Total number container positionings (sum of xm's) = 4
Total moving time (sum of y's) = 10

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3
-- 0 m 1 1 1 1 1 1 1 1 m 3 - - - -
- - - - 0 m m 2 2 2 2 2 2 m m 3 - -
- - - - - 0 m 1 1 1 m 3 - - - - -

```

Table E.26: CPPT model solution of the **4M4B.t** instance.

```

OBJECTIVE: 20
Total number container positionings (sum of xm's) = 6
Total moving time (sum of y's) = 14

0 m m 2 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
-- 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m 3 - - - - -
- - - - 0 m 1 1 1 1 1 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3 - - - - -
- - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m 3

```

Table E.27: CPPT model solution of the **4S.s** instance.

[illegible]

Table E.28: CPPT model solution of the 4C.s instance.

[illegible]

Table E.29: CPPT model solution of the 4M1A.s instance.

[illegible]

Table E.30: CPPT model solution of the 4M1B.s instance.

[illegible]

Table E.31: CPPT model solution of the 4M2A.s instance.

[illegible]

Table E.32: CPPT model solution of the 4M2B.s instance.

[illegible]

Table E.33: CPPT model solution of the 4M3A.s instance.

[illegible]

Table E.34: CPPT model solution of the 4M3B.s instance.

[illegible]

Table E.35: CPPT model solution of the 4M4A.s instance.

[illegible]

Table E.36: CPPT model solution of the 4M4B.s instance.

```

OBJECTIVE: 25
Total number container positionings (sum of xm's) = 7
Total moving time (sum of y's) = 18

0 m m 1 1 1 m 2 2 2 2 2 2 m 4 - - - - -
- - 0 m m 3 3 3 3 3 3 3 m m 4 - - - - -
- - - - 0 m 2 2 2 2 2 2 2 2 2 m 4 - - - -
- - - - - 0 m 2 2 2 m 1 1 1 1 1 m m 4 - -
- - - - - - 0 m m 1 1 1 1 1 1 1 m m 4

```

Table E.37: CPPT model solution of the 5S.t instance.

```

OBJECTIVE: 19
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 14

0 m m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 4
- - 0 m m 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 4 - -
- - - - 0 m 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - -
- - - - - 0 m 2 2 2 2 2 2 2 m 4 - - - - -
- - - - - - 0 m 2 2 2 m 4 - - - - - - - -

```

Table E.38: CPPT model solution of the 5C.t instance.

```

OBJECTIVE: 21
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 16

0 m m 1 1 1 1 1 1 m m 4 - - - - -
- - 0 m m 3 3 3 3 3 3 3 3 m m 4 - -
- - - - 0 m m 3 3 3 3 m m 4 - - - - -
- - - - - 0 m 2 2 2 2 2 2 2 2 m 4
- - - - - - 0 m 2 2 2 2 m 4 - - - -

```

Table E.39: CPPT model solution of the 5M1A.t instance.

```

OBJECTIVE: 21
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 16

0 m m 3 3 3 3 3 3 3 m m 4 - - - - -
- - 0 m m 1 1 1 1 1 1 1 m m 4 - - - -
- - - - 0 m 2 2 2 2 2 2 2 2 2 2 2 2 m 4
- - - - - 0 m m 1 1 1 m m 4 - - - - -
- - - - - - 0 m 2 2 2 2 2 2 2 m 4 - -

```

Table E.40: CPPT model solution of the 5M1B.t instance.

```

OBJECTIVE: 21
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 16

0 m 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - -
- - 0 m 2 2 2 2 2 2 2 m 4 - - - - -
- - - - 0 m m 3 3 3 3 3 3 3 3 3 m m 4 - -
- - - - - 0 m m 3 3 3 m m 4 - - - - -
- - - - - - 0 m m 1 1 1 1 1 1 1 m m 4

```

Table E.41: CPPT model solution of the 5M2A.t instance.

```

OBJECTIVE: 19
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 14

0 m 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - -
- - 0 m m 1 1 1 1 1 m m 4 - - - - -
- - - - 0 m 2 2 2 2 2 2 2 2 m 4 - - - -
- - - - - 0 m m 3 3 3 3 3 3 3 3 m m 4
- - - - - - 0 m 2 2 2 m 4 - - - - -

```

Table E.42: CPPT model solution of the 5M2B.t instance.

```

OBJECTIVE: 21
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 16

0 m 2 2 2 2 2 2 2 2 2 2 m 4 - - - - -
- - 0 m m 3 3 3 3 3 3 3 3 3 3 m m 4 - -
- - - - 0 m 2 2 2 2 2 m 4 - - - - -
- - - - - 0 m m 3 3 3 3 3 m m 4 - - - -
- - - - - - 0 m m 1 1 1 1 1 1 1 m m 4

```

Table E.43: CPPT model solution of the 5M3A.t instance.

```

OBJECTIVE: 19
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 14

0 m m 1 1 1 1 1 1 1 1 1 1 m m 4 - - - -
- - 0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4
- - - - 0 m m 1 1 1 m m 4 - - - - - - - -
- - - - - 0 m 2 2 2 2 2 2 2 2 2 m 4 - -
- - - - - - 0 m 2 2 2 m 4 - - - - - - - -

```

Table E.44: CPPT model solution of the 5M3B.t instance.

```

OBJECTIVE: 19
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 14

0 m m 1 1 1 1 1 1 1 1 1 1 m m 4 - - - -
- - 0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4
- - - - 0 m m 1 1 1 m m 4 - - - - - - - -
- - - - - 0 m 2 2 2 2 2 2 2 2 2 m 4 - -
- - - - - - 0 m 2 2 2 m 4 - - - - - - - -

```

Table E.45: CPPT model solution of the 5M4A.t instance.

```

OBJECTIVE: 19
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 14

0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - -
- - 0 m 2 2 2 2 2 2 2 2 2 2 m 4 - - - - - -
- - - - 0 m m 3 3 3 3 3 3 3 3 3 m m 4 - -
- - - - - 0 m 2 2 2 m 4 - - - - - - - -
- - - - - - 0 m m 1 1 1 1 1 1 m m 4

```

Table E.46: CPPT model solution of the 5M4B.t instance.

```

OBJECTIVE: 19
Total number container positionings (sum of xm's) = 5
Total moving time (sum of y's) = 14

0 m m 3 3 3 3 3 3 3 3 3 3 3 m m 4 - -
- - 0 m m 3 3 3 3 3 3 3 m m 4 - - - - - -
- - - - 0 m 2 2 2 2 2 2 2 2 2 2 2 m 4
- - - - - 0 m 2 2 2 2 2 2 m 4 - - - -
- - - - - - 0 m 2 m 4 - - - - - - - -

```

Table E.47: CPPT model solution of the 5M5A.t instance.

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | m | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | m | 4 | - | - | - | - |
| - | - | 0 | m | m | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | m | m | 4 |   |
| - | - | - | - | 0 | m | 2 | 2 | 2 | 2 | 2 | 2 | 2 | m | 4 | - | - | - | - |   |
| - | - | - | - | - | 0 | m | m | 1 | 1 | 1 | 1 | 1 | 1 | m | m | 4 | - | - |   |
| - | - | - | - | - | - | 0 | m | 2 | m | 4 | - | - | - | - | - | - | - | - |   |

Table E.48: CPPT model solution of the 5M5B.t instance.

[illegible]

Table E.49: CPPT model solution of the 5S.s instance.

[illegible]

Table E.50: CPPT model solution of the 5C.s instance.









```

OBJECTIVE: 27
Total number container positionings (sum of xm's) = 7
Total moving time (sum of y's) = 20

0 m m 3 3 3 3 3 3 3 3 3 m m 4 - - - - -
- - 0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - - -
- - - - 0 m m 1 1 1 1 1 1 1 1 1 1 1 1 m m 4 - -
- - - - - 0 m 2 2 2 2 2 2 2 m 4 - - - - -
- - - - - - 0 m m 1 1 1 1 1 1 1 m m 4 - - - -
- - - - - - - 0 m 2 2 m 3 3 3 3 3 3 3 m m 4

```

Table E.63: CPPT model solution of the 6M1.t instance.

```

OBJECTIVE: 26
Total number container positionings (sum of xm's) = 6
Total moving time (sum of y's) = 20

0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - -
- - 0 m 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - - -
- - - - 0 m m 1 1 1 1 1 1 1 1 1 1 1 1 m m 4 - -
- - - - - 0 m m 1 1 1 1 1 1 m m 4 - - - - -
- - - - - - 0 m m 3 3 3 3 3 3 3 3 3 3 m m 4
- - - - - - - 0 m m 3 3 3 3 m m 4 - - - - -

```

Table E.64: CPPT model solution of the 6M2.t instance.

```

OBJECTIVE: 22
Total number container positionings (sum of xm's) = 6
Total moving time (sum of y's) = 16

0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - -
- - 0 m 2 2 2 2 2 2 2 2 2 2 2 m 4 - - - - -
- - - - 0 m 2 2 2 2 2 2 m 4 - - - - -
- - - - - 0 m m 1 1 1 1 1 1 1 1 1 1 1 m m 4
- - - - - - 0 m m 1 1 1 1 1 1 m m 4 - - - -
- - - - - - - 0 m 2 2 m 4 - - - - -

```

Table E.65: CPPT model solution of the 6M3.t instance.

```

OBJECTIVE: 24
Total number container positionings (sum of xm's) = 6
Total moving time (sum of y's) = 18

0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4
- - 0 m 2 2 2 2 2 2 2 2 2 2 m 4 - - - - -
- - - - 0 m m 1 1 1 1 1 1 1 1 1 1 m m 4 - - - -
- - - - - 0 m 2 2 2 2 m 4 - - - - -
- - - - - - 0 m m 1 1 1 1 m m 4 - - - - -
- - - - - - - 0 m m 3 3 3 3 3 3 3 m m 4 - -

```

Table E.66: CPPT model solution of the 6M4.t instance.

```

0 m m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 4 - - -
- - 0 m m 1 1 1 1 1 1 1 1 1 1 m m 4 - - - - -
- - - - 0 m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - -
- - - - - 0 m 2 2 2 2 2 2 2 2 2 2 2 2 m 4 - -
- - - - - - 0 m 2 2 2 m 4 - - - - - - - -
- - - - - - - 0 m m 3 3 3 3 3 3 3 3 3 m m 4

```

Table E.67: CPPT model solution of the 6M5.t instance.

[illegible]

Table E.68: CPPT model solution of the 6M6.t instance.

[illegible]

Table E.69: CPPT model solution of the 6S.s instance.

[illegible]

Table E.70: CPPT model solution of the 6C.s instance.

[illegible]

Table E.71: CPPT model solution of the 6M1.s instance.

[illegible]

Table E.72: CPPT model solution of the **6M2.s** instance.

[illegible]

Table E.73: CPPT model solution of the 6M3.s instance.

[illegible]

Table E.74: CPPT model solution of the 6M4.s instance.

[illegible]

Table E.75: CPPT model solution of the 6M5.s instance.





```

OBJECTIVE: 40
Total number container positionings (sum of xm's) = 10
Total moving time (sum of y's) = 30

0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m m 5 - - - - -
- - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - -
- - - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - -
- - - - - - - 0 m m 3 3 3 3 m 4 4 4 4 4 4 4 4 4 4 4 4 4 m m 5
- - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - - - - - 0 m m 2 m 4 4 4 4 4 4 4 4 4 4 4 m m 5 - - -

```

Table E.79: CPPT model solution of the 8M1.t instance.

```

OBJECTIVE: 39
Total number container positionings (sum of xm's) = 11
Total moving time (sum of y's) = 28

0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m m m 5 - - - - -
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - -
- - - - - 0 m 1 1 1 1 1 1 m 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - 0 m m 2 2 2 2 m 4 4 4 4 4 4 4 4 4 4 4 4 m m 5 - -
- - - - - - - 0 m m 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 m 3 3 3 3 m 5
- - - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -

```

Table E.80: CPPT model solution of the 8M2.t instance.

```

OBJECTIVE: 42
Total number container positionings (sum of xm's) = 12
Total moving time (sum of y's) = 30

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - -
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - 0 m 1 1 1 1 1 1 1 1 1 m 2 2 2 2 2 2 m 4 m m 5 - - -
- - - - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 1 m m 5
- - - - - - - - 0 m m 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - - - - 0 m m m 4 4 4 4 4 4 4 4 4 4 m 3 3 m 5 - -

```

Table E.81: CPPT model solution of the 8M3.t instance.

```

OBJECTIVE: 37
Total number container positionings (sum of xm's) = 11
Total moving time (sum of y's) = 26

0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - -
- - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - - - - -
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - 0 m m 3 3 3 3 3 3 3 3 m 5 - - - - - - - - - - -
- - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - - - -
- - - - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 1 m m 5
- - - - - - - - 0 m 1 1 m 3 3 m 5 - - - - - - - - - - -
- - - - - - - - - 0 m 1 1 1 1 m 3 3 3 3 3 3 m 5 - - - -

```

Table E.82: CPPT model solution of the 8M4.t instance.

```

OBJECTIVE: 37
Total number container positionings (sum of xm's) = 11
Total moving time (sum of y's) = 26

0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5
- - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - -
- - - - 0 m m 2 m 4 4 4 4 4 4 4 4 m m 5 - - - - - - - - - -
- - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - - 0 m 1 1 1 m 3 3 m 5 - - - - - - - - - - -
- - - - - - - 0 m m 3 3 3 3 3 3 3 3 m 5 - - - - - - -
- - - - - - - - 0 m 1 1 1 1 1 1 m 3 3 3 3 3 3 m 5 - -
- - - - - - - - - 0 m m 3 3 3 m 5 - - - - - - - - - -

```

Table E.83: CPPT model solution of the 8M5.t instance.

```

OBJECTIVE: 37
Total number container positionings (sum of xm's) = 9
Total moving time (sum of y's) = 28

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - -
- - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - - - -
- - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - - -
- - - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 1 m m 5 - -
- - - - - - - 0 m m 3 3 3 3 m 5 - - - - - - - - - - -
- - - - - - - - 0 m 1 1 1 1 1 m m 5 - - - - - - - - -
- - - - - - - - - 0 m m 4 4 4 4 4 4 4 4 4 4 4 4 m m 5

```

Table E.84: CPPT model solution of the 8M6.t instance.

```

OBJECTIVE: 38
Total number container positionings (sum of xm's) = 10
Total moving time (sum of y's) = 28

0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - -
- - 0 m m 2 2 2 2 2 m 4 4 4 4 4 4 4 4 4 4 4 4 m m 5 - - - - - -
- - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - - - - -
- - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - -
- - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - -
- - - - - - - 0 m 1 1 1 1 1 1 1 1 m m 5 - - - - - - - - - -
- - - - - - - - 0 m m 3 3 m 5 - - - - - - - - - - - - - -
- - - - - - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 m 4 4 4 m m 5

```

Table E.85: CPPT model solution of the 8M7.t instance.

```

OBJECTIVE: 38
Total number container positionings (sum of xm's) = 10
Total moving time (sum of y's) = 28

0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - -
- - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - - -
- - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5
- - - - - 0 m m 3 m 4 4 4 4 4 4 4 4 4 4 4 4 4 4 m m 5 - - - -
- - - - - - 0 m 1 1 1 1 1 1 1 1 m m 5 - - - - - - - - - -
- - - - - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - -
- - - - - - - - 0 m m 3 3 m 4 4 4 4 4 4 m m 5 - - - - - -
- - - - - - - - - 0 m m 3 m 5 - - - - - - - - - - - -

```

Table E.86: CPPT model solution of the 8M8.t instance.

No feasible solution found within the time limit.

Table E.87: CPPT model solution of the 8S.s instance.

No feasible solution found within the time limit.

Table E.88: CPPT model solution of the 8C.s instance.

No feasible solution found within the time limit.

Table E.89: CPPT model solution of the 8M1.s instance.

No feasible solution found within the time limit.

Table E.90: CPPT model solution of the 8M2.s instance.

No feasible solution found within the time limit.

Table E.91: CPPT model solution of the 8M3.s instance.

No feasible solution found within the time limit.

Table E.92: CPPT model solution of the 8M4.s instance.

No feasible solution found within the time limit.

Table E.93: CPPT model solution of the 8M5.s instance.

No feasible solution found within the time limit.

Table E.94: CPPT model solution of the 8M6.s instance.

No feasible solution found within the time limit.

Table E.95: CPPT model solution of the 8M7.s instance.

No feasible solution found within the time limit.

Table E.96: CPPT model solution of the 8M8.s instance.

```

OBJECTIVE: 58
Total number container positionings (sum of xm's) = 18
Total moving time (sum of y's) = 40

0 m m 3 3 3 3 3 3 3 m 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 m m 5 - - - - -
- - - - 0 m 1 1 1 1 1 1 1 1 1 1 m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - - 0 m 1 1 1 1 1 m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - - - 0 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 m 5 - - - - -
- - - - - - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 4 4 4 4 4 4 4 m m 5 - - - - -
- - - - - - - - - - 0 m m 3 3 3 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - - - - - - - 0 m m m 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 m m 5 - - - - -
- - - - - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 m 3 3 3 3 3 3 3 3 3 3 3 3 m
- - - - - - - - - - - - - 0 m m 2 2 m m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
5 - - - -
3 3 3 m 5

```

Table E.97: CPPT model solution of the 10S.t instance.

```

OBJECTIVE: 72
Total number container positionings (sum of xm's) = 22
Total moving time (sum of y's) = 50

0 m m 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 m 1 1 1 1 1 1 1 1 1 1 m 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
- - - - 0 m m 2 2 2 2 m 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 m 3 3
- - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
- - - - - - - - 0 m m 2 2 2 2 2 2 2 2 2 2 2 m 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 m 3 3 3 m 1
- - - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m
- - - - - - - - - - 0 m m 2 2 2 m 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 m m 5 - -
- - - - - - - - - - 0 m 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 m m 5 - - - - -
- - - - - - - - - - - 0 m m m 4 4 4 4 4 4 4 4 4 4 4 4 m m 1 1 1 m m 5 - - - - -
- - - - - - - - - - - - 0 m m 2 2 2 2 m 1 1 1 m m 5 - - - - -
- - - - - - - - - - - - - 0 m m 3 3 m 5 - - - - -

3 3 3 3 3 3 3 3 3 m 4 4 4 4 4 m m 5
3 3 3 3 3 3 m 1 1 1 1 m m 5 - - - -
1 1 1 1 1 1 1 m m 5 - - - - -
1 1 1 m m 5 - - - - -
m 5 - - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

```

Table E.98: CPPT model solution of the 10C.t instance.













No feasible solution found within the time limit.

Table E.110: CPPT model solution of the 10C.s instance.

No feasible solution found within the time limit.

Table E.111: CPPT model solution of the 10M1.s instance.

No feasible solution found within the time limit.

Table E.112: CPPT model solution of the 10M2.s instance.

No feasible solution found within the time limit.

Table E.113: CPPT model solution of the 10M3.s instance.

No feasible solution found within the time limit.

Table E.114: CPPT model solution of the 10M4.s instance.

No feasible solution found within the time limit.

Table E.115: CPPT model solution of the 10M5.s instance.

No feasible solution found within the time limit.

Table E.116: CPPT model solution of the 10M6.s instance.

No feasible solution found within the time limit.

Table E.117: CPPT model solution of the 10M7.s instance.

No feasible solution found within the time limit.

Table E.118: CPPT model solution of the 10M8.s instance.

No feasible solution found within the time limit.

Table E.119: CPPT model solution of the 10M9.s instance.

No feasible solution found within the time limit.

Table E.120: CPPT model solution of the 10M10.s instance.



## Appendix F

# CPP and CPPT model solutions with extended time limit

### F.1 Scaled 10 containers subclass instances solved by the CPP model

OBJECTIVE: 54

Total number container positionings (sum of xp's) = 16

Total moving time (sum of xm's) = 38

Sequence

|       |  |   |   |   |   |
|-------|--|---|---|---|---|
| c     |  | 1 | 2 | 3 | 4 |
| ----- |  |   |   |   |   |
| 1     |  | 0 | 2 | 5 |   |
| 2     |  | 0 | 1 | 3 | 5 |
| 3     |  | 0 | 3 | 4 | 5 |
| 4     |  | 0 | 1 | 3 | 5 |
| 5     |  | 0 | 1 | 3 | 5 |
| 6     |  | 0 | 3 | 4 | 5 |
| 7     |  | 0 | 1 | 3 | 5 |
| 8     |  | 0 | 4 | 5 |   |
| 9     |  | 0 | 3 | 5 |   |
| 10    |  | 0 | 1 | 5 |   |

Positioning, storage, and moving times

|       |  |           |    |           |   |           |     |           |     |   |  |     |     |     |   |  |     |     |   |   |  |
|-------|--|-----------|----|-----------|---|-----------|-----|-----------|-----|---|--|-----|-----|-----|---|--|-----|-----|---|---|--|
| n     |  | Number: 1 |    | Number: 2 |   | Number: 3 |     | Number: 4 |     |   |  |     |     |     |   |  |     |     |   |   |  |
| c     |  | i :       | p  | s         | m |           | i : | p         | s   | m |  | i : | p   | s   | m |  | i : | p   | s | m |  |
| ----- |  |           |    |           |   |           |     |           |     |   |  |     |     |     |   |  |     |     |   |   |  |
| 1     |  | 0 :       | 1  | 0         | 2 |           | 2 : | 3         | 81  | 3 |  | 5 : | 87  | 0   | 0 |  |     |     |   |   |  |
| 2     |  | 0 :       | 5  | 0         | 1 |           | 1 : | 6         | 31  | 1 |  | 3 : | 38  | 60  | 1 |  | 5 : | 99  | 0 | 0 |  |
| 3     |  | 0 :       | 9  | 0         | 2 |           | 3 : | 11        | 21  | 1 |  | 4 : | 33  | 76  | 2 |  | 5 : | 111 | 0 | 0 |  |
| 4     |  | 0 :       | 13 | 0         | 1 |           | 1 : | 14        | 22  | 1 |  | 3 : | 37  | 85  | 1 |  | 5 : | 123 | 0 | 0 |  |
| 5     |  | 0 :       | 17 | 0         | 1 |           | 1 : | 18        | 17  | 1 |  | 3 : | 36  | 98  | 1 |  | 5 : | 135 | 0 | 0 |  |
| 6     |  | 0 :       | 21 | 0         | 2 |           | 3 : | 23        | 8   | 1 |  | 4 : | 32  | 113 | 2 |  | 5 : | 147 | 0 | 0 |  |
| 7     |  | 0 :       | 25 | 0         | 1 |           | 1 : | 26        | 8   | 1 |  | 3 : | 35  | 123 | 1 |  | 5 : | 159 | 0 | 0 |  |
| 8     |  | 0 :       | 29 | 0         | 3 |           | 4 : | 32        | 137 | 2 |  | 5 : | 171 | 0   | 0 |  |     |     |   |   |  |
| 9     |  | 0 :       | 33 | 0         | 2 |           | 3 : | 35        | 147 | 1 |  | 5 : | 183 | 0   | 0 |  |     |     |   |   |  |
| 10    |  | 0 :       | 37 | 0         | 1 |           | 1 : | 38        | 155 | 2 |  | 5 : | 195 | 0   | 0 |  |     |     |   |   |  |

Table F.1: CPP model solution of the 10S.s instance.

```

OBJECTIVE: 48
Total number container positionings (sum of xp's) = 12
Total moving time (sum of xm's) = 36

Sequence
c | 1 2 3 4
-----
1 | 0 1 3 5
2 | 0 1 3 5
3 | 0 3 5
4 | 0 2 5
5 | 0 3 5
6 | 0 4 5
7 | 0 3 5
8 | 0 4 5
9 | 0 1 5
10 | 0 1 5

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 | Number: 4 |
c | i : p s m | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 1 : 2 24 1 | 3 : 27 59 1 | 5 : 87 0 0 |
2 | 0 : 5 0 1 | 1 : 6 12 1 | 3 : 19 115 1 | 5 : 135 0 0 |
3 | 0 : 9 0 2 | 3 : 11 159 1 | 5 : 171 0 0 |
4 | 0 : 13 0 2 | 2 : 15 81 3 | 5 : 99 0 0 |
5 | 0 : 17 0 2 | 3 : 19 127 1 | 5 : 147 0 0 |
6 | 0 : 21 0 3 | 4 : 24 157 2 | 5 : 183 0 0 |
7 | 0 : 25 0 2 | 3 : 27 83 1 | 5 : 111 0 0 |
8 | 0 : 29 0 3 | 4 : 32 125 2 | 5 : 159 0 0 |
9 | 0 : 33 0 1 | 1 : 34 159 2 | 5 : 195 0 0 |
10 | 0 : 37 0 1 | 1 : 38 83 2 | 5 : 123 0 0 |

```

## F.2 Tight 10 containers subclass instances solved by the CPPT model

[illegible]







## Appendix G

# CPP and CPPT model solutions with relaxed lifo constraints

### G.1 Medium-scale class instances solved by the CPP model

```
OBJECTIVE: 18
Total number container positionings (sum of xp's) = 6
Total moving time (sum of xm's) = 12

Sequence
c | 1 2 3
-----
1 | 0 2 4
2 | 0 2 4
3 | 0 2 4
4 | 0 2 4
5 | 0 2 4
6 | 0 2 4

Positioning, storage, and moving times
n | Number: 1 | Number: 2 | Number: 3 |
c | i : p s m | i : p s m | i : p s m |
-----
1 | 0 : 1 0 1 | 2 : 2 51 1 | 4 : 54 0 0 |
2 | 0 : 3 0 1 | 2 : 4 55 1 | 4 : 60 0 0 |
3 | 0 : 5 0 1 | 2 : 6 59 1 | 4 : 66 0 0 |
4 | 0 : 7 0 1 | 2 : 8 63 1 | 4 : 72 0 0 |
5 | 0 : 9 0 1 | 2 : 10 67 1 | 4 : 78 0 0 |
6 | 0 : 11 0 1 | 2 : 12 71 1 | 4 : 84 0 0 |
```

Table G.1: CPP model solution of the 6S.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 2 4  
4 | 0 2 4  
5 | 0 2 4  
6 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 75 1 | 4 : 78 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 67 1 | 4 : 72 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 59 1 | 4 : 66 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 51 1 | 4 : 60 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 43 1 | 4 : 54 0 0 |  
6 | 0 : 11 0 1 | 2 : 12 35 1 | 4 : 48 0 0 |

Table G.2: CPP model solution of the 6C.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 2 4  
4 | 0 2 4  
5 | 0 2 4  
6 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 39 1 | 4 : 42 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 49 1 | 4 : 54 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 59 1 | 4 : 66 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 39 1 | 4 : 48 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 49 1 | 4 : 60 0 0 |  
6 | 0 : 11 0 1 | 2 : 12 59 1 | 4 : 72 0 0 |

Table G.3: CPP model solution of the 6M1.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 2 | 4 |

| Positioning, storage, and moving times |   |   |    |   |           |   |   |    |    |           |   |   |    |   |   |
|----------------------------------------|---|---|----|---|-----------|---|---|----|----|-----------|---|---|----|---|---|
| n   Number: 1                          |   |   |    |   | Number: 2 |   |   |    |    | Number: 3 |   |   |    |   |   |
| c                                      | i | p | s  | m | i         | p | s | m  | i  | p         | s | m |    |   |   |
| 1                                      | 0 | : | 1  | 0 | 1         | 2 | : | 2  | 60 | 1         | 4 | : | 63 | 0 | 0 |
| 2                                      | 0 | : | 3  | 0 | 1         | 2 | : | 4  | 40 | 1         | 4 | : | 45 | 0 | 0 |
| 3                                      | 0 | : | 5  | 0 | 1         | 2 | : | 6  | 62 | 1         | 4 | : | 69 | 0 | 0 |
| 4                                      | 0 | : | 7  | 0 | 1         | 2 | : | 8  | 42 | 1         | 4 | : | 51 | 0 | 0 |
| 5                                      | 0 | : | 9  | 0 | 1         | 2 | : | 10 | 64 | 1         | 4 | : | 75 | 0 | 0 |
| 6                                      | 0 | : | 11 | 0 | 1         | 2 | : | 12 | 44 | 1         | 4 | : | 57 | 0 | 0 |

Table G.4: CPP model solution of the 6M2.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 2 | 4 |

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s  | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1  | 0 | 1 | 2         | : | 2  | 60 | 1 | 4         | : | 63 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 2         | : | 4  | 46 | 1 | 4         | : | 51 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 2         | : | 6  | 32 | 1 | 4         | : | 39 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1 | 2         | : | 8  | 60 | 1 | 4         | : | 69 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1 | 2         | : | 10 | 46 | 1 | 4         | : | 57 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 2         | : | 12 | 32 | 1 | 4         | : | 45 | 0 | 0 |

Table G.5: CPP model solution of the 6M3.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 2 | 4 |

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s  | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1  | 0 | 1 | 2         | : | 2  | 69 | 1 | 4         | : | 72 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 2         | : | 4  | 43 | 1 | 4         | : | 48 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 2         | : | 6  | 53 | 1 | 4         | : | 60 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1 | 2         | : | 8  | 33 | 1 | 4         | : | 42 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1 | 2         | : | 10 | 43 | 1 | 4         | : | 54 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 2         | : | 12 | 53 | 1 | 4         | : | 66 | 0 | 0 |

Table G.6: CPP model solution of the 6M4.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 4 |
| 2 | 0 | 2 | 4 |
| 3 | 0 | 2 | 4 |
| 4 | 0 | 2 | 4 |
| 5 | 0 | 2 | 4 |
| 6 | 0 | 2 | 4 |

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|
| c | i         | p | s  | m |   | i         | p | s  | m  |   | i         | p | s  | m |   |
| 1 | 0         | : | 1  | 0 | 1 | 2         | : | 2  | 57 | 1 | 4         | : | 60 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 2         | : | 4  | 43 | 1 | 4         | : | 48 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 2         | : | 6  | 59 | 1 | 4         | : | 66 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1 | 2         | : | 8  | 45 | 1 | 4         | : | 54 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1 | 2         | : | 10 | 31 | 1 | 4         | : | 42 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 2         | : | 12 | 59 | 1 | 4         | : | 72 | 0 | 0 |

Table G.7: CPP model solution of the 6M5.s instance.

OBJECTIVE: 18  
Total number container positionings (sum of xp's) = 6  
Total moving time (sum of xm's) = 12

Sequence  
c | 1 2 3  
-----  
1 | 0 2 4  
2 | 0 2 4  
3 | 0 2 4  
4 | 0 2 4  
5 | 0 2 4  
6 | 0 2 4

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 2 : 2 69 1 | 4 : 72 0 0 |  
2 | 0 : 3 0 1 | 2 : 4 55 1 | 4 : 60 0 0 |  
3 | 0 : 5 0 1 | 2 : 6 41 1 | 4 : 48 0 0 |  
4 | 0 : 7 0 1 | 2 : 8 57 1 | 4 : 66 0 0 |  
5 | 0 : 9 0 1 | 2 : 10 43 1 | 4 : 54 0 0 |  
6 | 0 : 11 0 1 | 2 : 12 29 1 | 4 : 42 0 0 |

Table G.8: CPP model solution of the 6M6.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence  
c | 1 2 3  
-----  
1 | 0 1 5  
2 | 0 1 5  
3 | 0 1 5  
4 | 0 1 5  
5 | 0 1 5  
6 | 0 1 5  
7 | 0 1 5  
8 | 0 1 5

Positioning, storage, and moving times  
n | Number: 1 | Number: 2 | Number: 3 |  
c | i : p s m | i : p s m | i : p s m |  
-----  
1 | 0 : 1 0 1 | 1 : 2 71 2 | 5 : 75 0 0 |  
2 | 0 : 3 0 1 | 1 : 4 75 2 | 5 : 81 0 0 |  
3 | 0 : 5 0 1 | 1 : 6 79 2 | 5 : 87 0 0 |  
4 | 0 : 7 0 1 | 1 : 8 83 2 | 5 : 93 0 0 |  
5 | 0 : 9 0 1 | 1 : 10 87 2 | 5 : 99 0 0 |  
6 | 0 : 11 0 1 | 1 : 12 91 2 | 5 : 105 0 0 |  
7 | 0 : 13 0 1 | 1 : 14 95 2 | 5 : 111 0 0 |  
8 | 0 : 15 0 1 | 1 : 16 99 2 | 5 : 117 0 0 |

Table G.9: CPP model solution of the 8S.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   | Number: 2 |   |   |    | Number: 3 |   |   |   |     |   |   |
|---|-----------|---|----|---|-----------|---|---|----|-----------|---|---|---|-----|---|---|
| c | i         | p | s  | m | i         | p | s | m  | i         | p | s | m |     |   |   |
| 1 | 0         | : | 1  | 0 | 1         | 1 | : | 2  | 98        | 2 | 5 | : | 102 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1         | 1 | : | 4  | 90        | 2 | 5 | : | 96  | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1         | 1 | : | 6  | 82        | 2 | 5 | : | 90  | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1         | 1 | : | 8  | 74        | 2 | 5 | : | 84  | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1         | 1 | : | 10 | 66        | 2 | 5 | : | 78  | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1         | 1 | : | 12 | 58        | 2 | 5 | : | 72  | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1         | 1 | : | 14 | 50        | 2 | 5 | : | 66  | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1         | 1 | : | 16 | 42        | 2 | 5 | : | 60  | 0 | 0 |

Table G.10: CPP model solution of the 8C.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |     |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|-----|---|---|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p   | s | m |
| 1 | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 62 | 2 | 5         | : | 66  | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 1         | : | 4  | 78 | 2 | 5         | : | 84  | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 94 | 2 | 5         | : | 102 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1 | 1         | : | 8  | 62 | 2 | 5         | : | 72  | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 78 | 2 | 5         | : | 90  | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 1         | : | 12 | 94 | 2 | 5         | : | 108 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 62 | 2 | 5         | : | 78  | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1 | 1         | : | 16 | 78 | 2 | 5         | : | 96  | 0 | 0 |

Table G.11: CPP model solution of the 8M1.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |   |   |    | Number: 2 |   |   |   |   | Number: 3 |    |   |   |   |   |     |   |   |
|---|-----------|---|---|---|----|-----------|---|---|---|---|-----------|----|---|---|---|---|-----|---|---|
|   | c         | i | : | p | s  | m         | i | : | p | s | m         | i  | : | p | s | m |     |   |   |
| 1 | 1         | : | 0 | : | 1  | 0         | 1 | : | 1 | : | 2         | 86 | 2 | : | 5 | : | 90  | 0 | 0 |
| 2 | 1         | : | 0 | : | 3  | 0         | 1 | : | 1 | : | 4         | 60 | 2 | : | 5 | : | 66  | 0 | 0 |
| 3 | 1         | : | 0 | : | 5  | 0         | 1 | : | 1 | : | 6         | 88 | 2 | : | 5 | : | 96  | 0 | 0 |
| 4 | 1         | : | 0 | : | 7  | 0         | 1 | : | 1 | : | 8         | 62 | 2 | : | 5 | : | 72  | 0 | 0 |
| 5 | 1         | : | 0 | : | 9  | 0         | 1 | : | 1 | : | 10        | 90 | 2 | : | 5 | : | 102 | 0 | 0 |
| 6 | 1         | : | 0 | : | 11 | 0         | 1 | : | 1 | : | 12        | 64 | 2 | : | 5 | : | 78  | 0 | 0 |
| 7 | 1         | : | 0 | : | 13 | 0         | 1 | : | 1 | : | 14        | 92 | 2 | : | 5 | : | 108 | 0 | 0 |
| 8 | 1         | : | 0 | : | 15 | 0         | 1 | : | 1 | : | 16        | 66 | 2 | : | 5 | : | 84  | 0 | 0 |

Table G.12: CPP model solution of the 8M2.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |     |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|-----|---|---|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p   | s | m |
| 1 | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 65 | 2 | 5         | : | 69  | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 1         | : | 4  | 87 | 2 | 5         | : | 93  | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 55 | 2 | 5         | : | 63  | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1 | 1         | : | 8  | 77 | 2 | 5         | : | 87  | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 69 | 2 | 5         | : | 81  | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 1         | : | 12 | 91 | 2 | 5         | : | 105 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 59 | 2 | 5         | : | 75  | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1 | 1         | : | 16 | 81 | 2 | 5         | : | 99  | 0 | 0 |

Table G.13: CPP model solution of the 8M3.s instance.



OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |   |    |   |   |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|---|----|---|---|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p | s  | m |   |
| 1 | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 86 | 2 | 1         | : | 5 | 90 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1 | 1         | : | 4  | 60 | 2 | 1         | : | 5 | 66 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 70 | 2 | 1         | : | 5 | 78 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1 | 1         | : | 8  | 44 | 2 | 1         | : | 5 | 54 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 60 | 2 | 1         | : | 5 | 72 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1 | 1         | : | 12 | 82 | 2 | 1         | : | 5 | 96 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 44 | 2 | 1         | : | 5 | 60 | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1 | 1         | : | 16 | 66 | 2 | 1         | : | 5 | 84 | 0 | 0 |

Table G.14: CPP model solution of the 8M4.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   | Number: 2 |   |   |    | Number: 3 |   |   |   |    |   |   |
|---|-----------|---|----|---|-----------|---|---|----|-----------|---|---|---|----|---|---|
| c | i         | p | s  | m | i         | p | s | m  | i         | p | s | m |    |   |   |
| 1 | 0         | : | 1  | 0 | 1         | 1 | : | 2  | 92        | 2 | 5 | : | 96 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1         | 1 | : | 4  | 78        | 2 | 5 | : | 84 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1         | 1 | : | 6  | 52        | 2 | 5 | : | 60 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1         | 1 | : | 8  | 68        | 2 | 5 | : | 78 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1         | 1 | : | 10 | 42        | 2 | 5 | : | 54 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1         | 1 | : | 12 | 58        | 2 | 5 | : | 72 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1         | 1 | : | 14 | 74        | 2 | 5 | : | 90 | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1         | 1 | : | 16 | 48        | 2 | 5 | : | 66 | 0 | 0 |

Table G.15: CPP model solution of the 8M5.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |  |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|--|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p  | s | m |  |
| 1 | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 77 | 2 | 5         | : | 81 | 0 | 0 |  |
| 2 | 0         | : | 3  | 0 | 1 | 1         | : | 4  | 81 | 2 | 5         | : | 87 | 0 | 0 |  |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 55 | 2 | 5         | : | 63 | 0 | 0 |  |
| 4 | 0         | : | 7  | 0 | 1 | 1         | : | 8  | 65 | 2 | 5         | : | 75 | 0 | 0 |  |
| 5 | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 81 | 2 | 5         | : | 93 | 0 | 0 |  |
| 6 | 0         | : | 11 | 0 | 1 | 1         | : | 12 | 43 | 2 | 5         | : | 57 | 0 | 0 |  |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 53 | 2 | 5         | : | 69 | 0 | 0 |  |
| 8 | 0         | : | 15 | 0 | 1 | 1         | : | 16 | 81 | 2 | 5         | : | 99 | 0 | 0 |  |

Table G.16: CPP model solution of the 8M6.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   | Number: 2 |   |   |    | Number: 3 |   |   |   |    |   |   |
|---|-----------|---|----|---|-----------|---|---|----|-----------|---|---|---|----|---|---|
| c | i         | p | s  | m | i         | p | s | m  | i         | p | s | m |    |   |   |
| 1 | 0         | : | 1  | 0 | 1         | 1 | : | 2  | 80        | 2 | 5 | : | 84 | 0 | 0 |
| 2 | 0         | : | 3  | 0 | 1         | 1 | : | 4  | 66        | 2 | 5 | : | 72 | 0 | 0 |
| 3 | 0         | : | 5  | 0 | 1         | 1 | : | 6  | 52        | 2 | 5 | : | 60 | 0 | 0 |
| 4 | 0         | : | 7  | 0 | 1         | 1 | : | 8  | 80        | 2 | 5 | : | 90 | 0 | 0 |
| 5 | 0         | : | 9  | 0 | 1         | 1 | : | 10 | 66        | 2 | 5 | : | 78 | 0 | 0 |
| 6 | 0         | : | 11 | 0 | 1         | 1 | : | 12 | 52        | 2 | 5 | : | 66 | 0 | 0 |
| 7 | 0         | : | 13 | 0 | 1         | 1 | : | 14 | 38        | 2 | 5 | : | 54 | 0 | 0 |
| 8 | 0         | : | 15 | 0 | 1         | 1 | : | 16 | 78        | 2 | 5 | : | 96 | 0 | 0 |

Table G.17: CPP model solution of the 8M7.s instance.

OBJECTIVE: 32  
Total number container positionings (sum of xp's) = 8  
Total moving time (sum of xm's) = 24

Sequence

| c | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 5 |
| 2 | 0 | 1 | 5 |
| 3 | 0 | 1 | 5 |
| 4 | 0 | 1 | 5 |
| 5 | 0 | 1 | 5 |
| 6 | 0 | 1 | 5 |
| 7 | 0 | 1 | 5 |
| 8 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n | Number: 1 |   |    |   |   | Number: 2 |   |    |    |   | Number: 3 |   |    |   |   |  |
|---|-----------|---|----|---|---|-----------|---|----|----|---|-----------|---|----|---|---|--|
| c | i         | : | p  | s | m | i         | : | p  | s  | m | i         | : | p  | s | m |  |
| 1 | 0         | : | 1  | 0 | 1 | 1         | : | 2  | 83 | 2 | 5         | : | 87 | 0 | 0 |  |
| 2 | 0         | : | 3  | 0 | 1 | 1         | : | 4  | 63 | 2 | 5         | : | 69 | 0 | 0 |  |
| 3 | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 91 | 2 | 5         | : | 99 | 0 | 0 |  |
| 4 | 0         | : | 7  | 0 | 1 | 1         | : | 8  | 71 | 2 | 5         | : | 81 | 0 | 0 |  |
| 5 | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 51 | 2 | 5         | : | 63 | 0 | 0 |  |
| 6 | 0         | : | 11 | 0 | 1 | 1         | : | 12 | 79 | 2 | 5         | : | 93 | 0 | 0 |  |
| 7 | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 59 | 2 | 5         | : | 75 | 0 | 0 |  |
| 8 | 0         | : | 15 | 0 | 1 | 1         | : | 16 | 39 | 2 | 5         | : | 57 | 0 | 0 |  |

Table G.18: CPP model solution of the 8M8.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   | Number: 2 |   |   |    | Number: 3 |   |   |   |     |   |   |
|----|-----------|---|----|---|-----------|---|---|----|-----------|---|---|---|-----|---|---|
| c  | i         | p | s  | m | i         | p | s | m  | i         | p | s | m |     |   |   |
| 1  | 0         | : | 1  | 0 | 1         | 1 | : | 2  | 83        | 2 | 5 | : | 87  | 0 | 0 |
| 2  | 0         | : | 5  | 0 | 1         | 1 | : | 6  | 91        | 2 | 5 | : | 99  | 0 | 0 |
| 3  | 0         | : | 9  | 0 | 1         | 1 | : | 10 | 99        | 2 | 5 | : | 111 | 0 | 0 |
| 4  | 0         | : | 13 | 0 | 1         | 1 | : | 14 | 107       | 2 | 5 | : | 123 | 0 | 0 |
| 5  | 0         | : | 17 | 0 | 1         | 1 | : | 18 | 115       | 2 | 5 | : | 135 | 0 | 0 |
| 6  | 0         | : | 21 | 0 | 1         | 1 | : | 22 | 123       | 2 | 5 | : | 147 | 0 | 0 |
| 7  | 0         | : | 25 | 0 | 1         | 1 | : | 26 | 131       | 2 | 5 | : | 159 | 0 | 0 |
| 8  | 0         | : | 29 | 0 | 1         | 1 | : | 30 | 139       | 2 | 5 | : | 171 | 0 | 0 |
| 9  | 0         | : | 33 | 0 | 1         | 1 | : | 34 | 147       | 2 | 5 | : | 183 | 0 | 0 |
| 10 | 0         | : | 37 | 0 | 1         | 1 | : | 38 | 155       | 2 | 5 | : | 195 | 0 | 0 |

Table G.19: CPP model solution of the 10S.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| Number: 1 |   |   |    |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |   |     |   |   |
|-----------|---|---|----|---|-----------|---|---|---|----|-----------|---|---|---|---|-----|---|---|
| c         | i | p | s  | m | i         | p | s | m | i  | p         | s | m |   |   |     |   |   |
| 1         | 0 | : | 1  | 0 | 1         | : | 1 | : | 2  | 230       | 2 | : | 5 | : | 234 | 0 | 0 |
| 2         | 0 | : | 5  | 0 | 1         | : | 1 | : | 6  | 214       | 2 | : | 5 | : | 222 | 0 | 0 |
| 3         | 0 | : | 9  | 0 | 1         | : | 1 | : | 10 | 198       | 2 | : | 5 | : | 210 | 0 | 0 |
| 4         | 0 | : | 13 | 0 | 1         | : | 1 | : | 14 | 182       | 2 | : | 5 | : | 198 | 0 | 0 |
| 5         | 0 | : | 17 | 0 | 1         | : | 1 | : | 18 | 166       | 2 | : | 5 | : | 186 | 0 | 0 |
| 6         | 0 | : | 21 | 0 | 1         | : | 1 | : | 22 | 150       | 2 | : | 5 | : | 174 | 0 | 0 |
| 7         | 0 | : | 25 | 0 | 1         | : | 1 | : | 26 | 134       | 2 | : | 5 | : | 162 | 0 | 0 |
| 8         | 0 | : | 29 | 0 | 1         | : | 1 | : | 30 | 118       | 2 | : | 5 | : | 150 | 0 | 0 |
| 9         | 0 | : | 33 | 0 | 1         | : | 1 | : | 34 | 102       | 2 | : | 5 | : | 138 | 0 | 0 |
| 10        | 0 | : | 37 | 0 | 1         | : | 1 | : | 38 | 86        | 2 | : | 5 | : | 126 | 0 | 0 |

Table G.20: CPP model solution of the 10C.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   | Number: 2 |   |   |    | Number: 3 |   |   |   |    |     |   |   |
|----|-----------|---|----|---|-----------|---|---|----|-----------|---|---|---|----|-----|---|---|
| c  | i         | p | s  | m | i         | p | s | m  | i         | p | s | m |    |     |   |   |
| 1  | 0         | : | 0  | 1 | 1         | : | 2 | 83 | 2         | : | 5 | : | 87 | 0   | 0 |   |
| 2  | 0         | : | 5  | 0 | 1         | : | 1 | 6  | 127       | 2 | : | 5 | :  | 135 | 0 | 0 |
| 3  | 0         | : | 9  | 0 | 1         | : | 1 | 10 | 159       | 2 | : | 5 | :  | 171 | 0 | 0 |
| 4  | 0         | : | 13 | 0 | 1         | : | 1 | 14 | 83        | 2 | : | 5 | :  | 99  | 0 | 0 |
| 5  | 0         | : | 17 | 0 | 1         | : | 1 | 18 | 127       | 2 | : | 5 | :  | 147 | 0 | 0 |
| 6  | 0         | : | 21 | 0 | 1         | : | 1 | 22 | 159       | 2 | : | 5 | :  | 183 | 0 | 0 |
| 7  | 0         | : | 25 | 0 | 1         | : | 1 | 26 | 83        | 2 | : | 5 | :  | 111 | 0 | 0 |
| 8  | 0         | : | 29 | 0 | 1         | : | 1 | 30 | 127       | 2 | : | 5 | :  | 159 | 0 | 0 |
| 9  | 0         | : | 33 | 0 | 1         | : | 1 | 34 | 159       | 2 | : | 5 | :  | 195 | 0 | 0 |
| 10 | 0         | : | 37 | 0 | 1         | : | 1 | 38 | 83        | 2 | : | 5 | :  | 123 | 0 | 0 |

Table G.21: CPP model solution of the 10M1.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| Number: 1 |   |   |    |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |   |     |   |   |
|-----------|---|---|----|---|-----------|---|---|---|----|-----------|---|---|---|---|-----|---|---|
| c         | i | p | s  | m | i         | p | s | m | i  | p         | s | m |   |   |     |   |   |
| 1         | 0 | : | 1  | 0 | 1         | : | 1 | : | 2  | 143       | 2 | : | 5 | : | 147 | 0 | 0 |
| 2         | 0 | : | 5  | 0 | 1         | : | 1 | : | 6  | 79        | 2 | : | 5 | : | 87  | 0 | 0 |
| 3         | 0 | : | 9  | 0 | 1         | : | 1 | : | 10 | 147       | 2 | : | 5 | : | 159 | 0 | 0 |
| 4         | 0 | : | 13 | 0 | 1         | : | 1 | : | 14 | 83        | 2 | : | 5 | : | 99  | 0 | 0 |
| 5         | 0 | : | 17 | 0 | 1         | : | 1 | : | 18 | 151       | 2 | : | 5 | : | 171 | 0 | 0 |
| 6         | 0 | : | 21 | 0 | 1         | : | 1 | : | 22 | 87        | 2 | : | 5 | : | 111 | 0 | 0 |
| 7         | 0 | : | 25 | 0 | 1         | : | 1 | : | 26 | 155       | 2 | : | 5 | : | 183 | 0 | 0 |
| 8         | 0 | : | 29 | 0 | 1         | : | 1 | : | 30 | 91        | 2 | : | 5 | : | 123 | 0 | 0 |
| 9         | 0 | : | 33 | 0 | 1         | : | 1 | : | 34 | 159       | 2 | : | 5 | : | 195 | 0 | 0 |
| 10        | 0 | : | 37 | 0 | 1         | : | 1 | : | 38 | 95        | 2 | : | 5 | : | 135 | 0 | 0 |

Table G.22: CPP model solution of the 10M2.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   | Number: 2 |   |   |      | Number: 3 |   |     |   |       |   |   |
|----|-----------|---|----|---|-----------|---|---|------|-----------|---|-----|---|-------|---|---|
| c  | i         | p | s  | m | i         | p | s | m    | i         | p | s   | m |       |   |   |
| 1  | 0         | : | 0  | 1 | 1         | : | 1 | : 2  | 5         | : | 111 | 0 | 0     |   |   |
| 2  | 0         | : | 5  | 0 | 1         | : | 1 | : 6  | 107       | 2 | :   | 5 | : 171 | 0 | 0 |
| 3  | 0         | : | 9  | 0 | 1         | : | 1 | : 10 | 75        | 2 | :   | 5 | : 87  | 0 | 0 |
| 4  | 0         | : | 13 | 0 | 1         | : | 1 | : 14 | 131       | 2 | :   | 5 | : 147 | 0 | 0 |
| 5  | 0         | : | 17 | 0 | 1         | : | 1 | : 18 | 103       | 2 | :   | 5 | : 123 | 0 | 0 |
| 6  | 0         | : | 21 | 0 | 1         | : | 1 | : 22 | 159       | 2 | :   | 5 | : 183 | 0 | 0 |
| 7  | 0         | : | 25 | 0 | 1         | : | 1 | : 26 | 71        | 2 | :   | 5 | : 99  | 0 | 0 |
| 8  | 0         | : | 29 | 0 | 1         | : | 1 | : 30 | 127       | 2 | :   | 5 | : 159 | 0 | 0 |
| 9  | 0         | : | 33 | 0 | 1         | : | 1 | : 34 | 99        | 2 | :   | 5 | : 135 | 0 | 0 |
| 10 | 0         | : | 37 | 0 | 1         | : | 1 | : 38 | 155       | 2 | :   | 5 | : 195 | 0 | 0 |

Table G.23: CPP model solution of the 10M3.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| Number: 1 |   |   |    |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |   |     |   |   |
|-----------|---|---|----|---|-----------|---|---|---|----|-----------|---|---|---|---|-----|---|---|
| c         | i | p | s  | m | i         | p | s | m | i  | p         | s | m |   |   |     |   |   |
| 1         | 0 | : | 1  | 0 | 1         | : | 1 | : | 2  | 203       | 2 | : | 5 | : | 207 | 0 | 0 |
| 2         | 0 | : | 5  | 0 | 1         | : | 1 | : | 6  | 127       | 2 | : | 5 | : | 135 | 0 | 0 |
| 3         | 0 | : | 9  | 0 | 1         | : | 1 | : | 10 | 159       | 2 | : | 5 | : | 171 | 0 | 0 |
| 4         | 0 | : | 13 | 0 | 1         | : | 1 | : | 14 | 83        | 2 | : | 5 | : | 99  | 0 | 0 |
| 5         | 0 | : | 17 | 0 | 1         | : | 1 | : | 18 | 127       | 2 | : | 5 | : | 147 | 0 | 0 |
| 6         | 0 | : | 21 | 0 | 1         | : | 1 | : | 22 | 159       | 2 | : | 5 | : | 183 | 0 | 0 |
| 7         | 0 | : | 25 | 0 | 1         | : | 1 | : | 26 | 83        | 2 | : | 5 | : | 111 | 0 | 0 |
| 8         | 0 | : | 29 | 0 | 1         | : | 1 | : | 30 | 127       | 2 | : | 5 | : | 159 | 0 | 0 |
| 9         | 0 | : | 33 | 0 | 1         | : | 1 | : | 34 | 159       | 2 | : | 5 | : | 195 | 0 | 0 |
| 10        | 0 | : | 37 | 0 | 1         | : | 1 | : | 38 | 83        | 2 | : | 5 | : | 123 | 0 | 0 |

Table G.24: CPP model solution of the 10M4.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   |   | Number: 2 |   |    |     |   | Number: 3 |   |     |   |   |  |
|----|-----------|---|----|---|---|-----------|---|----|-----|---|-----------|---|-----|---|---|--|
| c  | i         | p | s  | m |   | i         | p | s  | m   |   | i         | p | s   | m |   |  |
| 1  | 0         | : | 0  | 1 |   | 1         | : | 2  | 179 | 2 | 5         | : | 183 | 0 | 0 |  |
| 2  | 0         | : | 5  | 0 | 1 | 1         | : | 6  | 151 | 2 | 5         | : | 159 | 0 | 0 |  |
| 3  | 0         | : | 9  | 0 | 1 | 1         | : | 10 | 123 | 2 | 5         | : | 135 | 0 | 0 |  |
| 4  | 0         | : | 13 | 0 | 1 | 1         | : | 14 | 95  | 2 | 5         | : | 111 | 0 | 0 |  |
| 5  | 0         | : | 17 | 0 | 1 | 1         | : | 18 | 67  | 2 | 5         | : | 87  | 0 | 0 |  |
| 6  | 0         | : | 21 | 0 | 1 | 1         | : | 22 | 75  | 2 | 5         | : | 99  | 0 | 0 |  |
| 7  | 0         | : | 25 | 0 | 1 | 1         | : | 26 | 95  | 2 | 5         | : | 123 | 0 | 0 |  |
| 8  | 0         | : | 29 | 0 | 1 | 1         | : | 30 | 115 | 2 | 5         | : | 147 | 0 | 0 |  |
| 9  | 0         | : | 33 | 0 | 1 | 1         | : | 34 | 135 | 2 | 5         | : | 171 | 0 | 0 |  |
| 10 | 0         | : | 37 | 0 | 1 | 1         | : | 38 | 155 | 2 | 5         | : | 195 | 0 | 0 |  |

Table G.25: CPP model solution of the 10M5.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| Number: 1 |   |   |    |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |   |     |   |   |
|-----------|---|---|----|---|-----------|---|---|---|----|-----------|---|---|---|---|-----|---|---|
| c         | i | p | s  | m | i         | p | s | m | i  | p         | s | m |   |   |     |   |   |
| 1         | 0 | : | 1  | 0 | 1         | : | 1 | : | 2  | 167       | 2 | : | 5 | : | 171 | 0 | 0 |
| 2         | 0 | : | 5  | 0 | 1         | : | 1 | : | 6  | 175       | 2 | : | 5 | : | 183 | 0 | 0 |
| 3         | 0 | : | 9  | 0 | 1         | : | 1 | : | 10 | 87        | 2 | : | 5 | : | 99  | 0 | 0 |
| 4         | 0 | : | 13 | 0 | 1         | : | 1 | : | 14 | 143       | 2 | : | 5 | : | 159 | 0 | 0 |
| 5         | 0 | : | 17 | 0 | 1         | : | 1 | : | 18 | 103       | 2 | : | 5 | : | 123 | 0 | 0 |
| 6         | 0 | : | 21 | 0 | 1         | : | 1 | : | 22 | 63        | 2 | : | 5 | : | 87  | 0 | 0 |
| 7         | 0 | : | 25 | 0 | 1         | : | 1 | : | 26 | 119       | 2 | : | 5 | : | 147 | 0 | 0 |
| 8         | 0 | : | 29 | 0 | 1         | : | 1 | : | 30 | 79        | 2 | : | 5 | : | 111 | 0 | 0 |
| 9         | 0 | : | 33 | 0 | 1         | : | 1 | : | 34 | 159       | 2 | : | 5 | : | 195 | 0 | 0 |
| 10        | 0 | : | 37 | 0 | 1         | : | 1 | : | 38 | 95        | 2 | : | 5 | : | 135 | 0 | 0 |

Table G.26: CPP model solution of the 10M6.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   |   | Number: 2 |   |   |     |     | Number: 3 |   |   |   |     |   |   |   |
|----|-----------|---|----|---|---|-----------|---|---|-----|-----|-----------|---|---|---|-----|---|---|---|
| c  | i         | p | s  | m |   | i         | p | s | m   |     | i         | p | s | m |     |   |   |   |
| 1  | 0         | : | 0  | 1 |   | 1         | : | 2 | 155 |     | 2         | : | 5 | : | 159 | 0 | 0 | : |
| 2  | 0         | : | 5  | 0 | 1 | :         | 1 | : | 6   | 187 | 2         | : | 5 | : | 195 | 0 | 0 | : |
| 3  | 0         | : | 9  | 0 | 1 | :         | 1 | : | 10  | 159 | 2         | : | 5 | : | 171 | 0 | 0 | : |
| 4  | 0         | : | 13 | 0 | 1 | :         | 1 | : | 14  | 83  | 2         | : | 5 | : | 99  | 0 | 0 | : |
| 5  | 0         | : | 17 | 0 | 1 | :         | 1 | : | 18  | 127 | 2         | : | 5 | : | 147 | 0 | 0 | : |
| 6  | 0         | : | 21 | 0 | 1 | :         | 1 | : | 22  | 87  | 2         | : | 5 | : | 111 | 0 | 0 | : |
| 7  | 0         | : | 25 | 0 | 1 | :         | 1 | : | 26  | 59  | 2         | : | 5 | : | 87  | 0 | 0 | : |
| 8  | 0         | : | 29 | 0 | 1 | :         | 1 | : | 30  | 91  | 2         | : | 5 | : | 123 | 0 | 0 | : |
| 9  | 0         | : | 33 | 0 | 1 | :         | 1 | : | 34  | 147 | 2         | : | 5 | : | 183 | 0 | 0 | : |
| 10 | 0         | : | 37 | 0 | 1 | :         | 1 | : | 38  | 95  | 2         | : | 5 | : | 135 | 0 | 0 | : |

Table G.27: CPP model solution of the 10M7.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| Number: 1 |   |   |    |   | Number: 2 |   |   |   |    | Number: 3 |   |   |   |   |     |   |   |
|-----------|---|---|----|---|-----------|---|---|---|----|-----------|---|---|---|---|-----|---|---|
| c         | i | p | s  | m | i         | p | s | m | i  | p         | s | m |   |   |     |   |   |
| 1         | 0 | : | 1  | 0 | 1         | : | 1 | : | 2  | 119       | 2 | : | 5 | : | 123 | 0 | 0 |
| 2         | 0 | : | 5  | 0 | 1         | : | 1 | : | 6  | 163       | 2 | : | 5 | : | 171 | 0 | 0 |
| 3         | 0 | : | 9  | 0 | 1         | : | 1 | : | 10 | 195       | 2 | : | 5 | : | 207 | 0 | 0 |
| 4         | 0 | : | 13 | 0 | 1         | : | 1 | : | 14 | 95        | 2 | : | 5 | : | 111 | 0 | 0 |
| 5         | 0 | : | 17 | 0 | 1         | : | 1 | : | 18 | 115       | 2 | : | 5 | : | 135 | 0 | 0 |
| 6         | 0 | : | 21 | 0 | 1         | : | 1 | : | 22 | 135       | 2 | : | 5 | : | 159 | 0 | 0 |
| 7         | 0 | : | 25 | 0 | 1         | : | 1 | : | 26 | 155       | 2 | : | 5 | : | 183 | 0 | 0 |
| 8         | 0 | : | 29 | 0 | 1         | : | 1 | : | 30 | 67        | 2 | : | 5 | : | 99  | 0 | 0 |
| 9         | 0 | : | 33 | 0 | 1         | : | 1 | : | 34 | 111       | 2 | : | 5 | : | 147 | 0 | 0 |
| 10        | 0 | : | 37 | 0 | 1         | : | 1 | : | 38 | 155       | 2 | : | 5 | : | 195 | 0 | 0 |

Table G.28: CPP model solution of the 10M8.s instance.

OBJECTIVE: 40  
Total number container positionings (sum of xp's) = 10  
Total moving time (sum of xm's) = 30

Sequence

| c  | 1 | 2 | 3 |
|----|---|---|---|
| 1  | 0 | 1 | 5 |
| 2  | 0 | 1 | 5 |
| 3  | 0 | 1 | 5 |
| 4  | 0 | 1 | 5 |
| 5  | 0 | 1 | 5 |
| 6  | 0 | 1 | 5 |
| 7  | 0 | 1 | 5 |
| 8  | 0 | 1 | 5 |
| 9  | 0 | 1 | 5 |
| 10 | 0 | 1 | 5 |

Positioning, storage, and moving times

| n  | Number: 1 |   |    |   |   | Number: 2 |   |   |     |     | Number: 3 |   |   |   |     |   |   |  |
|----|-----------|---|----|---|---|-----------|---|---|-----|-----|-----------|---|---|---|-----|---|---|--|
| c  | i         | p | s  | m |   | i         | p | s | m   |     | i         | p | s | m |     |   |   |  |
| 1  | 0         | : | 0  | 1 |   | 1         | : | 2 | 143 |     | 2         | : | 5 | : | 147 | 0 | 0 |  |
| 2  | 0         | : | 5  | 0 | 1 | :         | 1 | : | 6   | 187 | 2         | : | 5 | : | 195 | 0 | 0 |  |
| 3  | 0         | : | 9  | 0 | 1 | :         | 1 | : | 10  | 207 | 2         | : | 5 | : | 219 | 0 | 0 |  |
| 4  | 0         | : | 13 | 0 | 1 | :         | 1 | : | 14  | 107 | 2         | : | 5 | : | 123 | 0 | 0 |  |
| 5  | 0         | : | 17 | 0 | 1 | :         | 1 | : | 18  | 163 | 2         | : | 5 | : | 183 | 0 | 0 |  |
| 6  | 0         | : | 21 | 0 | 1 | :         | 1 | : | 22  | 111 | 2         | : | 5 | : | 135 | 0 | 0 |  |
| 7  | 0         | : | 25 | 0 | 1 | :         | 1 | : | 26  | 179 | 2         | : | 5 | : | 207 | 0 | 0 |  |
| 8  | 0         | : | 29 | 0 | 1 | :         | 1 | : | 30  | 127 | 2         | : | 5 | : | 159 | 0 | 0 |  |
| 9  | 0         | : | 33 | 0 | 1 | :         | 1 | : | 34  | 75  | 2         | : | 5 | : | 111 | 0 | 0 |  |
| 10 | 0         | : | 37 | 0 | 1 | :         | 1 | : | 38  | 131 | 2         | : | 5 | : | 171 | 0 | 0 |  |

Table G.29: CPP model solution of the 10M9.s instance.



| Sequence |  |   |   |   |
|----------|--|---|---|---|
| c        |  | 1 | 2 | 3 |
| 1        |  | 0 | 1 | 5 |
| 2        |  | 0 | 1 | 5 |
| 3        |  | 0 | 1 | 5 |
| 4        |  | 0 | 1 | 5 |
| 5        |  | 0 | 1 | 5 |
| 6        |  | 0 | 1 | 5 |
| 7        |  | 0 | 1 | 5 |
| 8        |  | 0 | 1 | 5 |
| 9        |  | 0 | 1 | 5 |
| 10       |  | 0 | 1 | 5 |

Table G.30: CPP model solution of the 10M10.s instance.

[illegible]

342

















No feasible solution found within the time limit.

Table G.51: CPPT model solution of the 10M1.s instance.

No feasible solution found within the time limit.

Table G.52: CPPT model solution of the 10M2.s instance.

No feasible solution found within the time limit.

Table G.53: CPPT model solution of the 10M3.s instance.

No feasible solution found within the time limit.

Table G.54: CPPT model solution of the 10M4.s instance.

No feasible solution found within the time limit.

Table G.55: CPPT model solution of the 10M5.s instance.

No feasible solution found within the time limit.

Table G.56: CPPT model solution of the 10M6.s instance.

No feasible solution found within the time limit.

Table G.57: CPPT model solution of the 10M7.s instance.

No feasible solution found within the time limit.

Table G.58: CPPT model solution of the 10M8.s instance.

No feasible solution found within the time limit.

Table G.59: CPPT model solution of the 10M9.s instance.

No feasible solution found within the time limit.

Table G.60: CPPT model solution of the 10M10.s instance.



# Appendix H

## Heuristic tuning results

### H.1 Small-scale class instances

| $C = 3$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.t    | 10.00 | 7/0      | 1/1.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 10.00 | 7/0      | 1/1.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| 3C.t    | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M1A.t  | 10.00 | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 10.00 | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M1B.t  | 10.00 | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M2A.t  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M2B.t  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M3A.t  | 8.00  | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8.00  | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M3B.t  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |

Table H.1: Tuning 1 on the 3 containers tight problem instances.

| $C = 3$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.t    | 10.00 | 7/0      | 1/1.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 10.00 | 7/0      | 1/1.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| 3C.t    | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M1A.t  | 10.00 | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M1B.t  | 10.00 | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M2A.t  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M2B.t  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M3A.t  | 8.00  | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8.00  | 6/0      | 1/1.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| 3M3B.t  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 8.00  | 6/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |

Table H.2: Tuning 2 on the 3 containers tight problem instances.

| $C = 3$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.s    | 18.00 | 8/6      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 0 | 2 | 6           | 33.33 | 66.67  | 0.1 |
| - no R  | 18.00 | 8/6      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 0 | 2 | 6           | 33.33 | 66.67  | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3C.s    | 12.00 | 6/4      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 12.00 | 6/4      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 12.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1A.s  | 18.00 | 7/5      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 18.00 | 7/5      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1B.s  | 18.00 | 7/5      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 1 | 6           | 16.67 | 83.33  | 0.3 |
| - no R  | 18.00 | 7/5      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 1/1.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2A.s  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2B.s  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3A.s  | 14.00 | 6/4      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 6/4      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3B.s  | 14.00 | 6/4      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 6/4      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 1/1.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.3: Tuning 0 on the 3 containers scaled problem instances.

| $C = 3$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.s    | 18.00 | 7/6      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 18.00 | 7/6      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3C.s    | 12.00 | 6/4      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 12.00 | 6/4      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 12.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1A.s  | 18.00 | 6/5      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 18.00 | 6/5      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1B.s  | 18.00 | 6/5      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 18.00 | 6/5      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 1/1.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2A.s  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2B.s  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3A.s  | 14.00 | 6/4      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/4      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3B.s  | 14.00 | 6/4      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/4      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 1/1.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.4: Tuning 1 on the 3 containers scaled problem instances.

| $C = 3$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 3S.s    | 18.00 | 7/6      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.1 |
| - no R  | 18.00 | 7/6      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 0 | 1 | 6           | 16.67 | 83.33  | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3C.s    | 12.00 | 6/4      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 12.00 | 6/4      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 12.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1A.s  | 18.00 | 6/5      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 6/5      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M1B.s  | 18.00 | 6/5      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 6/5      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 1/1.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2A.s  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M2B.s  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/5      | 2/3.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3A.s  | 14.00 | 6/4      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 6/4      | 2/5.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 3M3B.s  | 14.00 | 6/4      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 6/4      | 1/2.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 1 | 0 | 6           | 0.00  | 100.00 | 0.0 |
| - shift | 14.00 | -        | -      | 0/0.00 | 1/1.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.5: Tuning 2 on the 3 containers scaled problem instances.



| $C = 4$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.t    | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| 4C.t    | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M1A.t  | 14.00 | 9/0      | 0/0.00 | 0/0.00 | 2/2.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 14.00 | 9/0      | 0/0.00 | 0/0.00 | 2/2.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M1B.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| 4M2A.t  | 12.00 | 9/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 12.00 | 9/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M2B.t  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M3A.t  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M3B.t  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M4A.t  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M4B.t  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |

Table H.6: Tuning 1 on the 4 containers tight problem instances.

| $C = 4$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.t    | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| 4C.t    | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M1A.t  | 14.00 | 9/0      | 0/0.00 | 0/0.00 | 2/2.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 14.00 | 9/0      | 0/0.00 | 0/0.00 | 2/2.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M1B.t  | 14.00 | 9/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 14.00 | 9/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M2A.t  | 12.00 | 9/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 12.00 | 9/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| 4M2B.t  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M3A.t  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M3B.t  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M4A.t  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 12.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| 4M4B.t  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 10.00 | 8/0      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |

Table H.7: Tuning 2 on the 4 containers tight problem instances.

| $C = 4$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.s    | 26.00 | 10/9     | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - no R  | 26.00 | 10/9     | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4C.s    | 16.00 | 8/6      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 16.00 | 8/6      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 16.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1A.s  | 26.00 | 10/9     | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - no R  | 26.00 | 10/9     | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1B.s  | 26.00 | 11/9     | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 2 | 8           | 37.50 | 62.50  | 0.0 |
| - no R  | 26.00 | 11/9     | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 2 | 8           | 37.50 | 62.50  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2A.s  | 22.00 | 9/8      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 22.00 | 9/8      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2B.s  | 22.00 | 9/7      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 22.00 | 9/7      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3A.s  | 22.00 | 8/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 22.00 | 8/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3B.s  | 18.00 | 8/7      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 18.00 | 8/7      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4A.s  | 22.00 | 9/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 22.00 | 9/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4B.s  | 18.00 | 8/6      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 18.00 | 8/6      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.8: Tuning 0 on the 4 containers scaled problem instances.

| $C = 4$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.s    | 26.00 | 10/9     | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 26.00 | 10/9     | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4C.s    | 16.00 | 8/6      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 8/6      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 16.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1A.s  | 26.00 | 9/8      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 26.00 | 9/8      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1B.s  | 26.00 | 10/8     | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 26.00 | 10/8     | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2A.s  | 22.00 | 9/8      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 22.00 | 9/8      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2B.s  | 22.00 | 8/7      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 22.00 | 8/7      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3A.s  | 22.00 | 8/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 22.00 | 8/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3B.s  | 18.00 | 8/7      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 18.00 | 8/7      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4A.s  | 22.00 | 8/6      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 22.00 | 8/6      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4B.s  | 18.00 | 8/6      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 18.00 | 8/6      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.9: Tuning 1 on the 4 containers scaled problem instances.

| $C = 4$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 4S.s    | 26.00 | 10/9     | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - no R  | 26.00 | 10/9     | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 2 | 8           | 25.00 | 75.00  | 0.1 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4C.s    | 16.00 | 8/6      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 8/6      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 16.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1A.s  | 26.00 | 9/8      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - no R  | 26.00 | 9/8      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M1B.s  | 26.00 | 9/8      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 26.00 | 9/8      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 26.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2A.s  | 22.00 | 9/8      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.1 |
| - no R  | 22.00 | 9/8      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 1 | 1 | 8           | 12.50 | 87.50  | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M2B.s  | 22.00 | 8/7      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 22.00 | 8/7      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3A.s  | 22.00 | 8/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - no R  | 22.00 | 8/7      | 2/4.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M3B.s  | 18.00 | 8/7      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 8/7      | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4A.s  | 22.00 | 8/6      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 22.00 | 8/6      | 2/6.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 22.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 4M4B.s  | 18.00 | 8/6      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 8/6      | 1/2.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 2 | 0 | 8           | 0.00  | 100.00 | 0.0 |
| - shift | 18.00 | -        | -      | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.10: Tuning 2 on the 4 containers scaled problem instances.

| $C = 5$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.t    | 18.00 | 12/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 4        | 0 | 1 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00 | 1/1.00 | 0/0.00 | 4        | 0 | 1 | 2 | 10          | 20.00 | 80.00  | 0.0 |
| 5C.t    | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M1A.t  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M1B.t  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M2A.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M2B.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M3A.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M3B.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M4A.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M4B.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M5A.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M5B.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |

Table H.11: Tuning 1 on the 5 containers tight problem instances.

| $C = 5$ |       | Solution |        |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait   | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.t    | 18.00 | 12/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| 5C.t    | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M1A.t  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M1B.t  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 18.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M2A.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M2B.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M3A.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M3B.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M4A.t  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 16.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M4B.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M5A.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| 5M5B.t  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 14.00 | 10/0     | 0/0.00 | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |

Table H.12: Tuning 2 on the 5 containers tight problem instances.

| $C = 5$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.s    | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 2 | 10          | 20.00 | 80.00  | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5C.s    | 28.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 28.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 28.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1A.s  | 34.00 | 11/10    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 1 | 10          | 10.00 | 90.00  | 0.1 |
| - no R  | 34.00 | 11/10    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 1 | 10          | 10.00 | 90.00  | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1B.s  | 34.00 | 11/10    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 1 | 10          | 10.00 | 90.00  | 0.1 |
| - no R  | 34.00 | 11/10    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 1 | 10          | 10.00 | 90.00  | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2A.s  | 30.00 | 11/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.1 |
| - no R  | 30.00 | 11/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2B.s  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3A.s  | 30.00 | 11/10    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.1 |
| - no R  | 30.00 | 11/10    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3B.s  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4A.s  | 34.00 | 11/10    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.1 |
| - no R  | 34.00 | 11/10    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4B.s  | 30.00 | 11/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.1 |
| - no R  | 30.00 | 11/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 1 | 10          | 10.00 | 90.00  | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5A.s  | 26.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5B.s  | 26.00 | 10/8     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26.00 | 10/8     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.13: Tuning 0 on the 5 containers scaled problem instances.



| $C = 5$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.s    | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 1 | 2 | 10          | 20.00 | 80.00  | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5C.s    | 24.00 | 10/8     | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 24.00 | 10/8     | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 24.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1A.s  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1B.s  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2A.s  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2B.s  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3A.s  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3B.s  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4A.s  | 30.00 | 10/9     | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4B.s  | 26.00 | 10/9     | 2/4.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26.00 | 10/9     | 2/4.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5A.s  | 26.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 26.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5B.s  | 26.00 | 10/8     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 26.00 | 10/8     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.14: Tuning 1 on the 5 containers scaled problem instances.

| $C = 5$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 5S.s    | 34.00 | 12/11    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - no R  | 34.00 | 12/11    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 2 | 10          | 20.00 | 80.00  | 0.1 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5C.s    | 24.00 | 10/8     | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 24.00 | 10/8     | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 24.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1A.s  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M1B.s  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 34.00 | 10/9     | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2A.s  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M2B.s  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3A.s  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M3B.s  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 10/9     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4A.s  | 30.00 | 10/9     | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - no R  | 30.00 | 10/9     | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 30.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M4B.s  | 26.00 | 10/9     | 2/4.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26.00 | 10/9     | 2/4.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 2 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5A.s  | 26.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26.00 | 10/8     | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 5M5B.s  | 26.00 | 10/8     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.1 |
| - no R  | 26.00 | 10/8     | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 3 | 0 | 10          | 0.00  | 100.00 | 0.0 |
| - shift | 26.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.15: Tuning 2 on the 5 containers scaled problem instances.

## H.2 Medium-scale class instances

| $C = 6$ |       | Solution |        |         |        |         | Criteria |   |   |   | Performance |        |        |     |
|---------|-------|----------|--------|---------|--------|---------|----------|---|---|---|-------------|--------|--------|-----|
| T 1     | T     | #M       | Wait   | Early   | Late   | R       | 0        | 1 | 2 | 3 | LB          | GAP    | QUAL   | CPU |
| 6S.t    | 22.00 | 15/0     | 0/0.00 | 0/0.00  | 1/1.00 | 0/0.00  | 4        | 0 | 2 | 3 | 12          | 25.00  | 75.00  | 0.1 |
| - no R  | 22.00 | 15/0     | 0/0.00 | 0/0.00  | 1/1.00 | 0/0.00  | 4        | 0 | 2 | 3 | 12          | 25.00  | 75.00  | 0.0 |
| 6C.t    | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M1.t   | 22.00 | 13/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 4        | 0 | 2 | 1 | 12          | 8.33   | 91.67  | 0.1 |
| - no R  | 22.00 | 13/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 4        | 0 | 2 | 1 | 12          | 8.33   | 91.67  | 0.0 |
| 6M2.t   | 20.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 20.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M3.t   | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M4.t   | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M5.t   | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M6.t   | 14.00 | 10/0     | 0/0.00 | 1/14.00 | 0/0.00 | 1/16.67 | 2        | 0 | 3 | 0 | 12          | -16.67 | 116.67 | 0.0 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 1/1.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |

Table H.16: Tuning 1 on the 6 containers tight problem instances.

| $C = 6$ |       | Solution |        |         |        |         | Criteria |   |   |   | Performance |        |        |     |
|---------|-------|----------|--------|---------|--------|---------|----------|---|---|---|-------------|--------|--------|-----|
| T 2     | T     | #M       | Wait   | Early   | Late   | R       | 0        | 1 | 2 | 3 | LB          | GAP    | QUAL   | CPU |
| 6S.t    | 22.00 | 15/0     | 0/0.00 | 0/0.00  | 1/1.00 | 0/0.00  | 4        | 0 | 2 | 3 | 12          | 25.00  | 75.00  | 0.1 |
| - no R  | 22.00 | 15/0     | 0/0.00 | 0/0.00  | 1/1.00 | 0/0.00  | 4        | 0 | 2 | 3 | 12          | 25.00  | 75.00  | 0.1 |
| 6C.t    | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M1.t   | 22.00 | 13/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 4        | 0 | 2 | 1 | 12          | 8.33   | 91.67  | 0.1 |
| - no R  | 22.00 | 13/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 4        | 0 | 2 | 1 | 12          | 8.33   | 91.67  | 0.0 |
| 6M2.t   | 20.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 20.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M3.t   | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 2/2.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M4.t   | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M5.t   | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 0/0.00 | 0/0.00  | 3        | 0 | 3 | 0 | 12          | 0.00   | 100.00 | 0.0 |
| 6M6.t   | 14.00 | 10/0     | 0/0.00 | 1/14.00 | 0/0.00 | 1/16.67 | 2        | 0 | 3 | 0 | 12          | -16.67 | 116.67 | 0.1 |
| - no R  | 18.00 | 12/0     | 0/0.00 | 0/0.00  | 1/1.00 | 0/0.00  | 2        | 0 | 4 | 0 | 12          | 0.00   | 100.00 | 0.0 |

Table H.17: Tuning 2 on the 6 containers tight problem instances.

| $C = 6$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 6S.s    | 42.00 | 15/14    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.1 |
| - no R  | 42.00 | 15/14    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.0 |
| - shift | 42.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6C.s    | 36.00 | 12/10    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 36.00 | 12/10    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 36.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M1.s   | 42.00 | 14/13    | 4/16.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 2 | 12          | 16.67 | 83.33  | 0.1 |
| - no R  | 42.00 | 14/13    | 4/16.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 2 | 12          | 16.67 | 83.33  | 0.0 |
| - shift | 42.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M2.s   | 38.00 | 14/13    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 2 | 12          | 16.67 | 83.33  | 0.1 |
| - no R  | 38.00 | 14/13    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 2 | 12          | 16.67 | 83.33  | 0.0 |
| - shift | 38.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M3.s   | 34.00 | 12/11    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34.00 | 12/11    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M4.s   | 38.00 | 13/12    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.1 |
| - no R  | 38.00 | 13/12    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.0 |
| - shift | 38.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M5.s   | 34.00 | 13/12    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.1 |
| - no R  | 34.00 | 13/12    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M6.s   | 32.00 | 12/10    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 12/10    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.18: Tuning 0 on the 6 containers scaled problem instances.

| $C = 6$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 6S.s    | 42.00 | 15/14    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.1 |
| - no R  | 42.00 | 15/14    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.0 |
| - shift | 42.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6C.s    | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M1.s   | 42.00 | 13/12    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.1 |
| - no R  | 42.00 | 13/12    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.0 |
| - shift | 42.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M2.s   | 38.00 | 12/11    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 38.00 | 12/11    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 38.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M3.s   | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M4.s   | 34.00 | 12/11    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - no R  | 34.00 | 12/11    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M5.s   | 34.00 | 12/11    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - no R  | 34.00 | 12/11    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M6.s   | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - no R  | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.19: Tuning 1 on the 6 containers scaled problem instances.

| $C = 6$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 6S.s    | 42.00 | 15/14    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.1 |
| - no R  | 42.00 | 15/14    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 3 | 12          | 25.00 | 75.00  | 0.0 |
| - shift | 42.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6C.s    | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M1.s   | 42.00 | 13/12    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.1 |
| - no R  | 42.00 | 13/12    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 2 | 1 | 12          | 8.33  | 91.67  | 0.0 |
| - shift | 42.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M2.s   | 38.00 | 12/11    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 38.00 | 12/11    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 38.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M3.s   | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34.00 | 12/11    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M4.s   | 34.00 | 12/11    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34.00 | 12/11    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M5.s   | 34.00 | 12/11    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 34.00 | 12/11    | 3/10.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 3 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 34.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 6M6.s   | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 12/10    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 2        | 0 | 4 | 0 | 12          | 0.00  | 100.00 | 0.0 |
| - shift | 32.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.20: Tuning 2 on the 6 containers scaled problem instances.

| $C = 8$ |       | Solution |        |        |         |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|---------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait   | Early  | Late    | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.t    | 34.00 | 18/0     | 1/1.00 | 1/1.00 | 5/12.00 | 0/0.00 | 4        | 0 | 2 | 4 | 16          | 12.50 | 87.50  | 0.1 |
| - no R  | 34.00 | 18/0     | 1/1.00 | 1/1.00 | 5/12.00 | 0/0.00 | 4        | 0 | 2 | 4 | 16          | 12.50 | 87.50  | 0.1 |
| 8C.t    | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 6/15.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 6/15.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M1.t   | 34.00 | 17/0     | 2/2.00 | 0/0.00 | 7/12.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.1 |
| - no R  | 34.00 | 17/0     | 2/2.00 | 0/0.00 | 7/12.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| 8M2.t   | 32.00 | 16/0     | 1/1.00 | 0/0.00 | 2/2.00  | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 32.00 | 16/0     | 1/1.00 | 0/0.00 | 2/2.00  | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M3.t   | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 5/13.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 5/13.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M4.t   | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M5.t   | 30.00 | 16/0     | 1/1.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 1/1.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M6.t   | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 7/15.00 | 0/0.00 | 4        | 0 | 3 | 1 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 7/15.00 | 0/0.00 | 4        | 0 | 3 | 1 | 16          | 0.00  | 100.00 | 0.0 |
| 8M7.t   | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 8/20.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 8/20.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M8.t   | 28.00 | 16/0     | 0/0.00 | 0/0.00 | 3/5.00  | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 28.00 | 16/0     | 0/0.00 | 0/0.00 | 3/5.00  | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |

Table H.21: Tuning 1 on the 8 containers tight problem instances.



| $C = 8$ |       | Solution |        |        |         |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|--------|--------|---------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait   | Early  | Late    | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.t    | 34.00 | 20/0     | 1/1.00 | 0/0.00 | 4/9.00  | 0/0.00 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.1 |
| - no R  | 34.00 | 20/0     | 1/1.00 | 0/0.00 | 4/9.00  | 0/0.00 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| 8C.t    | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 6/15.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 6/15.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M1.t   | 32.00 | 16/0     | 2/2.00 | 0/0.00 | 2/4.00  | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 32.00 | 16/0     | 2/2.00 | 0/0.00 | 2/4.00  | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M2.t   | 32.00 | 16/0     | 1/1.00 | 0/0.00 | 2/2.00  | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 16/0     | 1/1.00 | 0/0.00 | 2/2.00  | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M3.t   | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 5/13.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 5/13.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M4.t   | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M5.t   | 30.00 | 16/0     | 1/1.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 1/1.00 | 0/0.00 | 8/17.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M6.t   | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 7/15.00 | 0/0.00 | 4        | 0 | 3 | 1 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 30.00 | 16/0     | 0/0.00 | 0/0.00 | 7/15.00 | 0/0.00 | 4        | 0 | 3 | 1 | 16          | 0.00  | 100.00 | 0.0 |
| 8M7.t   | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 8/20.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 32.00 | 16/0     | 0/0.00 | 0/0.00 | 8/20.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| 8M8.t   | 28.00 | 16/0     | 0/0.00 | 0/0.00 | 3/5.00  | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 28.00 | 16/0     | 0/0.00 | 0/0.00 | 3/5.00  | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |

Table H.22: Tuning 2 on the 8 containers tight problem instances.

| $C = 8$ |     | Solution |        |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|---------|-----|----------|--------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| T 0     | T   | #M       | Wait   | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.s    | 65  | 20/19    | 5/34.0 | 0/0.0 | 1/8.0  | 0/0.0 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - no R  | 65  | 20/19    | 5/34.0 | 0/0.0 | 1/8.0  | 0/0.0 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - shift | 65  | -        | -      | 0/0.0 | 1/8.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8C.s    | 55  | 16/14    | 5/28.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 55  | 16/14    | 5/28.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 55  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M1.s   | 65  | 17/16    | 6/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - no R  | 65  | 17/16    | 6/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - shift | 65  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M2.s   | 69  | 18/17    | 5/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - no R  | 69  | 18/17    | 5/38.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - shift | 69  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M3.s   | 61  | 16/15    | 6/36.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61  | 16/15    | 6/36.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M4.s   | 61  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M5.s   | 57  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 57  | 16/15    | 5/32.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M6.s   | 65  | 18/17    | 5/36.0 | 0/0.0 | 2/10.0 | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - no R  | 65  | 18/17    | 5/36.0 | 0/0.0 | 2/10.0 | 0/0.0 | 5        | 0 | 3 | 2 | 16          | 12.50 | 87.50  | 0.0 |
| - shift | 67  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M7.s   | 55  | 16/15    | 5/26.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 55  | 16/15    | 5/26.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 55  | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M8.s   | 55  | 16/14    | 5/24.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 55  | 16/14    | 5/24.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 55. | -        | -      | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.23: Tuning 0 on the 8 containers scaled problem instances.

| $C = 8$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.s    | 73.00 | 20/19    | 4/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 6        | 0 | 2 | 4 | 16          | 25.00 | 75.00  | 0.1 |
| - no R  | 73.00 | 20/19    | 4/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 6        | 0 | 2 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - shift | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8C.s    | 57.00 | 16/14    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 57.00 | 16/14    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M1.s   | 65.00 | 17/16    | 6/38.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - no R  | 65.00 | 17/16    | 6/38.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - shift | 65.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M2.s   | 61.00 | 16/15    | 5/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61.00 | 16/15    | 5/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M3.s   | 61.00 | 16/15    | 6/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61.00 | 16/15    | 6/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M4.s   | 57.00 | 16/15    | 5/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 57.00 | 16/15    | 5/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M5.s   | 57.00 | 16/15    | 4/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 57.00 | 16/15    | 4/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M6.s   | 55.00 | 17/16    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - no R  | 55.00 | 17/16    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - shift | 55.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M7.s   | 61.00 | 16/15    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61.00 | 16/15    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M8.s   | 53.00 | 16/14    | 5/24.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 53.00 | 16/14    | 5/24.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 53.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.24: Tuning 1 on the 8 containers scaled problem instances.

| $C = 8$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|---------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2     | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 8S.s    | 65.00 | 20/19    | 4/22.00 | 0/0.00 | 1/4.00 | 0/0.00 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.1 |
| - no R  | 65.00 | 20/19    | 4/22.00 | 0/0.00 | 1/4.00 | 0/0.00 | 5        | 0 | 3 | 4 | 16          | 25.00 | 75.00  | 0.0 |
| - shift | 65.00 | -        | -       | 0/0.00 | 1/4.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8C.s    | 57.00 | 16/14    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 57.00 | 16/14    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M1.s   | 61.00 | 16/15    | 6/46.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61.00 | 16/15    | 6/46.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M2.s   | 61.00 | 16/15    | 5/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61.00 | 16/15    | 5/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M3.s   | 61.00 | 16/15    | 6/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 61.00 | 16/15    | 6/32.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M4.s   | 57.00 | 16/15    | 5/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 57.00 | 16/15    | 5/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M5.s   | 57.00 | 16/15    | 4/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 57.00 | 16/15    | 4/22.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 57.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M6.s   | 55.00 | 17/16    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.1 |
| - no R  | 55.00 | 17/16    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 1 | 16          | 6.25  | 93.75  | 0.0 |
| - shift | 55.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M7.s   | 61.00 | 16/15    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - no R  | 61.00 | 16/15    | 4/20.00 | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 4 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 61.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 8M8.s   | 53.00 | 16/14    | 5/24.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.1 |
| - no R  | 53.00 | 16/14    | 5/24.00 | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 5 | 0 | 16          | 0.00  | 100.00 | 0.0 |
| - shift | 53.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.25: Tuning 2 on the 8 containers scaled problem instances.

| $C = 10$ |       | Solution |         |         |         |         | Criteria |   |   |   | Performance |       |        |     |
|----------|-------|----------|---------|---------|---------|---------|----------|---|---|---|-------------|-------|--------|-----|
| T 1      | T     | #M       | Wait    | Early   | Late    | R       | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.t    | 40.00 | 24/0     | 3/6.00  | 1/2.00  | 3/9.00  | 0/0.00  | 6        | 0 | 3 | 5 | 20          | 20.00 | 80.00  | 0.1 |
| - no R   | 40.00 | 24/0     | 3/6.00  | 1/2.00  | 3/9.00  | 0/0.00  | 6        | 0 | 3 | 5 | 20          | 20.00 | 80.00  | 0.1 |
| 10C.t    | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M1.t   | 39.00 | 20/0     | 3/5.00  | 1/41.00 | 3/5.00  | 1/10.00 | 6        | 0 | 3 | 2 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 42.00 | 22/0     | 3/5.00  | 0/0.00  | 5/8.00  | 0/0.00  | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| 10M2.t   | 42.00 | 21/0     | 3/3.00  | 0/0.00  | 4/4.00  | 0/0.00  | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 42.00 | 21/0     | 3/3.00  | 0/0.00  | 4/4.00  | 0/0.00  | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M3.t   | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M4.t   | 38.00 | 21/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 38.00 | 21/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M5.t   | 42.00 | 22/0     | 4/11.00 | 0/0.00  | 4/13.00 | 0/0.00  | 7        | 0 | 3 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 42.00 | 22/0     | 4/11.00 | 0/0.00  | 4/13.00 | 0/0.00  | 7        | 0 | 3 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| 10M6.t   | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M7.t   | 38.00 | 22/0     | 2/6.00  | 0/0.00  | 1/2.00  | 0/0.00  | 3        | 0 | 7 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - no R   | 38.00 | 22/0     | 2/6.00  | 0/0.00  | 1/2.00  | 0/0.00  | 3        | 0 | 7 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| 10M8.t   | 40.00 | 21/0     | 0/0.00  | 0/0.00  | 1/1.00  | 0/0.00  | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 40.00 | 21/0     | 0/0.00  | 0/0.00  | 1/1.00  | 0/0.00  | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M9.t   | 38.00 | 21/0     | 0/0.00  | 0/0.00  | 1/1.00  | 0/0.00  | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 38.00 | 21/0     | 0/0.00  | 0/0.00  | 1/1.00  | 0/0.00  | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M10.t  | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00  | 0/0.00  | 0/0.00  | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |

Table H.26: Tuning 1 on the 10 containers tight problem instances.

| $C = 10$ |       | Solution |         |        |         |        | Criteria |   |   |   | Performance |       |        |     |
|----------|-------|----------|---------|--------|---------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2      | T     | #M       | Wait    | Early  | Late    | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.t    | 40.00 | 24/0     | 3/6.00  | 1/2.00 | 3/9.00  | 0/0.00 | 6        | 0 | 3 | 5 | 20          | 20.00 | 80.00  | 0.1 |
| - no R   | 40.00 | 24/0     | 3/6.00  | 1/2.00 | 3/9.00  | 0/0.00 | 6        | 0 | 3 | 5 | 20          | 20.00 | 80.00  | 0.0 |
| 10C.t    | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M1.t   | 38.00 | 20/0     | 1/2.00  | 0/0.00 | 1/2.00  | 0/0.00 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 1/2.00  | 0/0.00 | 1/2.00  | 0/0.00 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M2.t   | 42.00 | 21/0     | 3/3.00  | 0/0.00 | 4/4.00  | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 42.00 | 21/0     | 3/3.00  | 0/0.00 | 4/4.00  | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M3.t   | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 5        | 0 | 5 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M4.t   | 38.00 | 21/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 38.00 | 21/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M5.t   | 40.00 | 22/0     | 4/15.00 | 0/0.00 | 4/10.00 | 0/0.00 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 40.00 | 22/0     | 4/15.00 | 0/0.00 | 4/10.00 | 0/0.00 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| 10M6.t   | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M7.t   | 40.00 | 20/0     | 1/2.00  | 0/0.00 | 1/2.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 40.00 | 20/0     | 1/2.00  | 0/0.00 | 1/2.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M8.t   | 38.00 | 21/0     | 0/0.00  | 0/0.00 | 1/1.00  | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 38.00 | 21/0     | 0/0.00  | 0/0.00 | 1/1.00  | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| 10M9.t   | 40.00 | 20/0     | 0/0.00  | 0/0.00 | 1/1.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 40.00 | 20/0     | 0/0.00  | 0/0.00 | 1/1.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| 10M10.t  | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 38.00 | 20/0     | 0/0.00  | 0/0.00 | 0/0.00  | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |

Table H.27: Tuning 2 on the 10 containers tight problem instances.

| $C = 10$ |    | Solution |       |       |        |       | Criteria |   |   |   | Performance |       |        |     |
|----------|----|----------|-------|-------|--------|-------|----------|---|---|---|-------------|-------|--------|-----|
| T 0      | T  | #M       | Wait  | Early | Late   | R     | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.s    | 93 | 27/26    | 3/6.0 | 0/0.0 | 1/14.0 | 0/0.0 | 7        | 0 | 3 | 7 | 20          | 35.00 | 65.00  | 0.1 |
| - no R   | 93 | 27/26    | 3/6.0 | 0/0.0 | 1/14.0 | 0/0.0 | 7        | 0 | 3 | 7 | 20          | 35.00 | 65.00  | 0.0 |
| - shift  | 95 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10C.s    | 69 | 20/18    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 69 | 20/18    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 69 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M1.s   | 85 | 25/23    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 5 | 20          | 25.00 | 75.00  | 0.1 |
| - no R   | 85 | 25/23    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 5 | 20          | 25.00 | 75.00  | 0.0 |
| - shift  | 85 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M2.s   | 81 | 23/22    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R   | 81 | 23/22    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 81 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M3.s   | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R   | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 77 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M4.s   | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R   | 77 | 23/22    | 2/4.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 77 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M5.s   | 79 | 22/21    | 2/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 79 | 22/21    | 2/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 6        | 0 | 4 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift  | 79 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M6.s   | 79 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 79 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift  | 79 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M7.s   | 73 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 73 | 22/21    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift  | 73 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M8.s   | 77 | 21/20    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 77 | 21/20    | 1/2.0 | 0/0.0 | 0/0.0  | 0/0.0 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 77 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M9.s   | 73 | 21/20    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 73 | 21/20    | 0/0.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 73 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M10.s  | 73 | 22/20    | 3/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 73 | 22/20    | 3/6.0 | 0/0.0 | 0/0.0  | 0/0.0 | 4        | 0 | 6 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift  | 73 | -        | -     | 0/0.0 | 0/0.0  | -     | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.28: Tuning 0 on the 10 containers scaled problem instances.

| $C = 10$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|----------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 1      | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.s    | 85.00 | 26/25    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 6        | 0 | 4 | 6 | 20          | 30.00 | 70.00  | 0.1 |
| - no R   | 85.00 | 26/25    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 6        | 0 | 4 | 6 | 20          | 30.00 | 70.00  | 0.1 |
| - shift  | 85.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10C.s    | 73.00 | 20/18    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 73.00 | 20/18    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M1.s   | 85.00 | 22/21    | 4/14.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 85.00 | 22/21    | 4/14.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift  | 85.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M2.s   | 85.00 | 23/22    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - no R   | 85.00 | 23/22    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 85.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M3.s   | 77.00 | 21/20    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 77.00 | 21/20    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M4.s   | 77.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 77.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M5.s   | 79.00 | 23/22    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R   | 79.00 | 23/22    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 79.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M6.s   | 77.00 | 20/19    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 77.00 | 20/19    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M7.s   | 73.00 | 22/21    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 2 | 20          | 10.00 | 90.00  | 0.1 |
| - no R   | 73.00 | 22/21    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 2 | 20          | 10.00 | 90.00  | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M8.s   | 77.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 77.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M9.s   | 73.00 | 21/20    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 73.00 | 21/20    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M10.s  | 73.00 | 20/18    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 73.00 | 20/18    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.29: Tuning 1 on the 10 containers scaled problem instances.



| $C = 10$ |       | Solution |         |        |        |        | Criteria |   |   |   | Performance |       |        |     |
|----------|-------|----------|---------|--------|--------|--------|----------|---|---|---|-------------|-------|--------|-----|
| T 2      | T     | #M       | Wait    | Early  | Late   | R      | 0        | 1 | 2 | 3 | LB          | GAP   | QUAL   | CPU |
| 10S.s    | 81.00 | 26/25    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 6 | 20          | 30.00 | 70.00  | 0.1 |
| - no R   | 81.00 | 26/25    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 6 | 20          | 30.00 | 70.00  | 0.1 |
| - shift  | 81.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10C.s    | 73.00 | 20/18    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 73.00 | 20/18    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M1.s   | 85.00 | 21/20    | 4/14.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 85.00 | 21/20    | 4/14.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 85.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M2.s   | 89.00 | 23/22    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 6 | 2 | 20          | 15.00 | 85.00  | 0.1 |
| - no R   | 89.00 | 23/22    | 3/12.00 | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 6 | 2 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 89.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M3.s   | 85.00 | 21/20    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 85.00 | 21/20    | 2/6.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 85.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M4.s   | 77.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 77.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M5.s   | 79.00 | 23/22    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.1 |
| - no R   | 79.00 | 23/22    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 5        | 0 | 5 | 3 | 20          | 15.00 | 85.00  | 0.0 |
| - shift  | 79.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M6.s   | 77.00 | 20/19    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - no R   | 77.00 | 20/19    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M7.s   | 73.00 | 21/20    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.1 |
| - no R   | 73.00 | 21/20    | 0/0.00  | 0/0.00 | 0/0.00 | 0/0.00 | 3        | 0 | 7 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M8.s   | 73.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - no R   | 73.00 | 21/20    | 1/2.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 1 | 20          | 5.00  | 95.00  | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M9.s   | 77.00 | 20/19    | 2/4.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 77.00 | 20/19    | 2/4.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 77.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |
| 10M10.s  | 73.00 | 20/18    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.1 |
| - no R   | 73.00 | 20/18    | 3/8.00  | 0/0.00 | 0/0.00 | 0/0.00 | 4        | 0 | 6 | 0 | 20          | 0.00  | 100.00 | 0.0 |
| - shift  | 73.00 | -        | -       | 0/0.00 | 0/0.00 | -      | -        | - | - | - | -           | -     | -      | 0.0 |

Table H.30: Tuning 2 on the 10 containers scaled problem instances.

## Heuristic solutions

| c   i : p s m   i : p s m   i : p s m   i : p s m |  |     |   |   |   |  |     |   |   |   |  |     |                |
|---------------------------------------------------|--|-----|---|---|---|--|-----|---|---|---|--|-----|----------------|
| 1                                                 |  | A : | 1 | 0 | 1 |  | 1 : | 2 | 4 | 1 |  | D : | 7              |
| 2                                                 |  | A : | 2 | 0 | 2 |  | 2 : | 4 | 4 | 2 |  | D : | 10             |
| 3                                                 |  | A : | 3 | 1 | 2 |  | 2 : | 6 | 1 | 1 |  | 1 : | 8 2 1   D : 11 |

| c l i : p s m   i : p s m   i : p s m |  |     |   |   |   |  |     |   |   |   |  |        |
|---------------------------------------|--|-----|---|---|---|--|-----|---|---|---|--|--------|
| 1                                     |  | A : | 1 | 0 | 1 |  | 1 : | 2 | 7 | 1 |  | D : 10 |
| 2                                     |  | A : | 2 | 0 | 1 |  | 1 : | 3 | 5 | 1 |  | D : 9  |
| 3                                     |  | A : | 3 | 0 | 2 |  | 2 : | 5 | 1 | 2 |  | D : 8  |

| c   i : p s m   i : p s m   i : p s m |  |     |   |   |   |  |     |   |   |   |  |     |    |
|---------------------------------------|--|-----|---|---|---|--|-----|---|---|---|--|-----|----|
| 1                                     |  | A : | 1 | 0 | 1 |  | 1 : | 2 | 4 | 1 |  | D : | 7  |
| 2                                     |  | A : | 2 | 0 | 2 |  | 2 : | 4 | 6 | 2 |  | D : | 12 |
| 3                                     |  | A : | 3 | 1 | 2 |  | 2 : | 6 | 2 | 2 |  | D : | 10 |

383

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 6 | 1 | D : | 9  |  |  |
| 2                                     | A : | 3 | 0 | 2 | 2 : | 5 | 6 | 2 | D : | 13 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 2 : | 7 | 2 | 2 | D : | 11 |  |  |

Table I.4: Heuristic solution of the 3M1B.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |   |  |  |
|---------------------------------------|-----|---|---|---|-----|---|---|---|-----|---|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 4 | 1 | D : | 7 |  |  |
| 2                                     | A : | 2 | 0 | 1 | 1 : | 3 | 2 | 1 | D : | 6 |  |  |
| 3                                     | A : | 3 | 0 | 2 | 2 : | 5 | 2 | 2 | D : | 9 |  |  |

Table I.5: Heuristic solution of the 3M2A.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |   |  |  |
|---------------------------------------|-----|---|---|---|-----|---|---|---|-----|---|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 6 | 1 | D : | 9 |  |  |
| 2                                     | A : | 2 | 0 | 1 | 1 : | 3 | 2 | 1 | D : | 6 |  |  |
| 3                                     | A : | 3 | 0 | 2 | 2 : | 5 | 1 | 2 | D : | 8 |  |  |

Table I.6: Heuristic solution of the 3M2B.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 7 | 1 | D : | 10 |  |  |
| 2                                     | A : | 2 | 0 | 2 | 2 : | 4 | 6 | 2 | D : | 12 |  |  |
| 3                                     | A : | 3 | 1 | 1 | 1 : | 5 | 2 | 1 | D : | 8  |  |  |

Table I.7: Heuristic solution of the 3M3A.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 7 | 1 | D : | 10 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 2 : | 5 | 5 | 2 | D : | 12 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 1 : | 6 | 1 | 1 | D : | 8  |  |  |

Table I.8: Heuristic solution of the 3M3B.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |   |            |
|---------------------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|---|------------|
| 1                                                 | A : | 1 | 0 | 1 | 1 : | 2 | 18 | 1 | D : | 21 |   |            |
| 2                                                 | A : | 2 | 1 | 2 | 2 : | 5 | 20 | 2 | D : | 27 |   |            |
| 3                                                 | A : | 3 | 4 | 2 | 2 : | 9 | 14 | 1 | 1 : | 24 | 8 | 1   D : 33 |

Table I.9: Heuristic solution of the 3S.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 27 | 1 | D : | 30 |  |  |
| 2                                     | A : | 2 | 1 | 1 | 1 : | 4 | 22 | 1 | D : | 27 |  |  |
| 3                                     | A : | 3 | 2 | 2 | 2 : | 7 | 15 | 2 | D : | 24 |  |  |

Table I.10: Heuristic solution of the 3C.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 18 | 1 | D : | 21 |  |  |
| 2                                     | A : | 2 | 1 | 2 | 2 : | 5 | 29 | 2 | D : | 36 |  |  |
| 3                                     | A : | 3 | 4 | 2 | 2 : | 9 | 19 | 2 | D : | 30 |  |  |

Table I.11: Heuristic solution of the 3M1A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 24 | 1 | D : | 27 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 2 : | 5 | 30 | 2 | D : | 37 |  |  |
| 3                                     | A : | 5 | 2 | 2 | 2 : | 9 | 22 | 2 | D : | 33 |  |  |

Table I.12: Heuristic solution of the 3M1B.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 18 | 1 | D : | 21 |  |  |
| 2                                     | A : | 2 | 1 | 1 | 1 : | 4 | 13 | 1 | D : | 18 |  |  |
| 3                                     | A : | 3 | 2 | 2 | 2 : | 7 | 18 | 2 | D : | 27 |  |  |

Table I.13: Heuristic solution of the 3M2A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 24 | 1 | D : | 27 |  |  |
| 2                                     | A : | 2 | 1 | 1 | 1 : | 4 | 13 | 1 | D : | 18 |  |  |
| 3                                     | A : | 3 | 2 | 2 | 2 : | 7 | 15 | 2 | D : | 24 |  |  |

Table I.14: Heuristic solution of the 3M2B.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 27 | 1 | D : | 30 |  |  |
| 2                                     | A : | 2 | 1 | 2 | 2 : | 5 | 29 | 2 | D : | 36 |  |  |
| 3                                     | A : | 3 | 4 | 1 | 1 : | 8 | 15 | 1 | D : | 24 |  |  |

Table I.15: Heuristic solution of the 3M3A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 27 | 1 | D : | 30 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 2 : | 5 | 27 | 2 | D : | 34 |  |  |
| 3                                     | A : | 5 | 2 | 1 | 1 : | 8 | 15 | 1 | D : | 24 |  |  |

Table I.16: Heuristic solution of the 3M3B.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |   |            |
|---------------------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|---|------------|
| 1                                                 | A : | 1 | 0 | 1 | 1 : | 2 | 9 | 1 | D : | 12 |   |            |
| 2                                                 | A : | 3 | 0 | 1 | 1 : | 4 | 5 | 1 | 2 : | 10 | 2 | 2   D : 14 |
| 3                                                 | A : | 5 | 0 | 2 | 2 : | 7 | 8 | 2 | D : | 17 |   |            |
| 4                                                 | A : | 7 | 0 | 2 | 2 : | 9 | 5 | 1 | 1 : | 15 | 2 | 1   D : 18 |

Table I.17: Heuristic solution of the 4S.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 15 | 1 | D : | 18 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 1 : | 4 | 11 | 1 | D : | 16 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 1 : | 6 | 7  | 1 | D : | 14 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 2 : | 9 | 1  | 2 | D : | 12 |  |  |

Table I.18: Heuristic solution of the 4C.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |   |            |
|---------------------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|---|------------|
| 1                                                 | A : | 1 | 0 | 1 | 1 : | 2 | 8 | 1 | D : | 11 |   |            |
| 2                                                 | A : | 3 | 0 | 2 | 2 : | 5 | 9 | 2 | D : | 16 |   |            |
| 3                                                 | A : | 5 | 0 | 2 | 2 : | 7 | 5 | 2 | D : | 14 |   |            |
| 4                                                 | A : | 7 | 0 | 2 | 2 : | 9 | 2 | 1 | 1 : | 12 | 4 | 1   D : 17 |

Table I.19: Heuristic solution of the 4M1A.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |   |            |
|---------------------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|---|------------|
| 1                                                 | A : | 1 | 0 | 1 | 1 : | 2 | 7 | 1 | D : | 10 |   |            |
| 2                                                 | A : | 3 | 0 | 2 | 2 : | 5 | 8 | 2 | D : | 15 |   |            |
| 3                                                 | A : | 5 | 0 | 2 | 2 : | 7 | 5 | 1 | 1 : | 13 | 2 | 1   D : 16 |
| 4                                                 | A : | 7 | 0 | 2 | 2 : | 9 | 1 | 2 | D : | 12 |   |            |

Table I.20: Heuristic solution of the 4M1B.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |   |            |
|---------------------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|---|------------|
| 1                                                 | A : | 1 | 0 | 1 | 1 : | 2 | 9 | 1 | D : | 12 |   |            |
| 2                                                 | A : | 3 | 0 | 1 | 1 : | 4 | 5 | 1 | D : | 10 |   |            |
| 3                                                 | A : | 5 | 0 | 2 | 2 : | 7 | 6 | 2 | D : | 15 |   |            |
| 4                                                 | A : | 7 | 0 | 2 | 2 : | 9 | 3 | 1 | 1 : | 13 | 2 | 1   D : 16 |

Table I.21: Heuristic solution of the 4M2A.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 11 | 1 | D : | 14 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 1 : | 4 | 5  | 1 | D : | 10 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 2 : | 7 | 7  | 2 | D : | 16 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 2 : | 9 | 1  | 2 | D : | 12 |  |  |

Table I.22: Heuristic solution of the 4M2B.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |   |   |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|---|---|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 1 : | 2 | 9 | 1 | D : | 12 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 2 : | 5 | 9 | 2 | D : | 16 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 1 : | 6 | 3 | 1 | D : | 10 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 2 : | 9 | 3 | 2 | D : | 14 |  |  |

Table I.23: Heuristic solution of the 4M3A.t instance.

| c | i | p | s | m | i | p | s  | m | i | p  | s | m | i |
|---|---|---|---|---|---|---|----|---|---|----|---|---|---|
| 1 | A | 1 | 0 | 1 | 1 | 2 | 12 | 1 | D | 15 |   |   |   |
| 2 | A | 3 | 0 | 1 | 1 | 4 | 6  | 1 | D | 11 |   |   |   |
| 3 | A | 5 | 0 | 1 | 1 | 6 | 3  | 1 | D | 10 |   |   |   |
| 4 | A | 7 | 0 | 2 | 2 | 9 | 2  | 2 | D | 13 |   |   |   |

Table I.24: Heuristic solution of the 4M3B.t instance.

| c | i | p | s | m | i | p | s | m | i | p  | s | m | i |
|---|---|---|---|---|---|---|---|---|---|----|---|---|---|
| 1 | A | 1 | 0 | 1 | 1 | 2 | 9 | 1 | D | 12 |   |   |   |
| 2 | A | 3 | 0 | 2 | 2 | 5 | 9 | 2 | D | 16 |   |   |   |
| 3 | A | 5 | 0 | 2 | 2 | 7 | 5 | 2 | D | 14 |   |   |   |
| 4 | A | 7 | 0 | 1 | 1 | 8 | 1 | 1 | D | 10 |   |   |   |

Table I.25: Heuristic solution of the 4M4A.t instance.

| c | i | p | s | m | i | p | s  | m | i | p  | s | m |
|---|---|---|---|---|---|---|----|---|---|----|---|---|
| 1 | A | 1 | 0 | 1 | 1 | 2 | 15 | 1 | D | 18 |   |   |
| 2 | A | 3 | 0 | 1 | 1 | 4 | 9  | 1 | D | 14 |   |   |
| 3 | A | 5 | 0 | 2 | 2 | 7 | 7  | 2 | D | 16 |   |   |
| 4 | A | 7 | 0 | 1 | 1 | 8 | 3  | 1 | D | 12 |   |   |

Table I.26: Heuristic solution of the 4M4B.t instance.

| c | i | p | s | m | i | p  | s  | m | i | p  | s  | m | i | p  | s | m | i |
|---|---|---|---|---|---|----|----|---|---|----|----|---|---|----|---|---|---|
| 1 | A | 1 | 0 | 1 | 1 | 2  | 33 | 1 | D | 36 |    |   |   |    |   |   |   |
| 2 | A | 3 | 0 | 1 | 1 | 4  | 8  | 1 | D | 13 | 27 | 2 | D | 42 |   |   |   |
| 3 | A | 5 | 0 | 2 | 1 | 7  | 39 | 2 | D | 48 |    |   |   |    |   |   |   |
| 4 | A | 7 | 2 | 2 | 2 | 11 | 33 | 1 | 1 | 45 | 8  | 1 | D | 54 |   |   |   |

Table I.27: Heuristic solution of the 4S.s instance.

| c | i : | p | s | m | i : | p | s  | m | i : | p  | s | m |
|---|-----|---|---|---|-----|---|----|---|-----|----|---|---|
| 1 | A : | 1 | 0 | 1 | 1 : | 2 | 51 | 1 | D : | 54 |   |   |
| 2 | A : | 3 | 0 | 1 | 1 : | 4 | 43 | 1 | D : | 48 |   |   |
| 3 | A : | 5 | 0 | 1 | 1 : | 6 | 35 | 1 | D : | 42 |   |   |
| 4 | A : | 7 | 0 | 2 | 2 : | 9 | 25 | 2 | D : | 36 |   |   |

Table I.28: Heuristic solution of the 4C.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |   |   |         |    |    |         |    |    |         |    |  |  |
|---------------------------------------------------|---|---|---------|----|----|---------|----|----|---------|----|--|--|
| 1   A :                                           | 1 | 0 | 1   1 : | 2  | 30 | 1   D : | 33 |    |         |    |  |  |
| 2   A :                                           | 3 | 0 | 2   2 : | 5  | 38 | 2   D : | 45 |    |         |    |  |  |
| 3   A :                                           | 5 | 2 | 2   2 : | 9  | 28 | 2   D : | 39 |    |         |    |  |  |
| 4   A :                                           | 7 | 4 | 2   2 : | 13 | 22 | 1   1 : | 36 | 14 | 1   D : | 51 |  |  |

Table I.29: Heuristic solution of the 4M1A.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |   |   |         |    |    |         |    |   |         |    |  |  |
|---------------------------------------------------|---|---|---------|----|----|---------|----|---|---------|----|--|--|
| 1   A :                                           | 1 | 0 | 1   1 : | 2  | 27 | 1   D : | 30 |   |         |    |  |  |
| 2   A :                                           | 3 | 0 | 2   2 : | 5  | 35 | 2   D : | 42 |   |         |    |  |  |
| 3   A :                                           | 5 | 2 | 2   2 : | 9  | 29 | 1   1 : | 39 | 8 | 1   D : | 48 |  |  |
| 4   A :                                           | 7 | 4 | 2   2 : | 13 | 21 | 2   D : | 36 |   |         |    |  |  |

Table I.30: Heuristic solution of the 4M1B.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |   |   |         |    |    |         |    |   |         |    |  |  |
|---------------------------------------------------|---|---|---------|----|----|---------|----|---|---------|----|--|--|
| 1   A :                                           | 1 | 0 | 1   1 : | 2  | 33 | 1   D : | 36 |   |         |    |  |  |
| 2   A :                                           | 3 | 0 | 1   1 : | 4  | 25 | 1   D : | 30 |   |         |    |  |  |
| 3   A :                                           | 5 | 0 | 2   2 : | 7  | 33 | 2   D : | 42 |   |         |    |  |  |
| 4   A :                                           | 7 | 2 | 2   2 : | 11 | 27 | 1   1 : | 39 | 8 | 1   D : | 48 |  |  |

Table I.31: Heuristic solution of the 4M2A.s instance.

| c   i : p s m   i : p s m   i : p s m |   |   |         |    |    |         |    |  |  |  |  |  |
|---------------------------------------|---|---|---------|----|----|---------|----|--|--|--|--|--|
| 1   A :                               | 1 | 0 | 1   1 : | 2  | 39 | 1   D : | 42 |  |  |  |  |  |
| 2   A :                               | 3 | 0 | 1   1 : | 4  | 25 | 1   D : | 30 |  |  |  |  |  |
| 3   A :                               | 5 | 0 | 2   2 : | 7  | 39 | 2   D : | 48 |  |  |  |  |  |
| 4   A :                               | 7 | 2 | 2   2 : | 11 | 23 | 2   D : | 36 |  |  |  |  |  |

Table I.32: Heuristic solution of the 4M2B.s instance.

| c   i : p s m   i : p s m   i : p s m |   |   |         |    |    |         |    |  |  |  |  |  |
|---------------------------------------|---|---|---------|----|----|---------|----|--|--|--|--|--|
| 1   A :                               | 1 | 0 | 1   1 : | 2  | 33 | 1   D : | 36 |  |  |  |  |  |
| 2   A :                               | 3 | 0 | 2   2 : | 5  | 41 | 2   D : | 48 |  |  |  |  |  |
| 3   A :                               | 5 | 2 | 1   1 : | 8  | 21 | 1   D : | 30 |  |  |  |  |  |
| 4   A :                               | 7 | 2 | 2   2 : | 11 | 29 | 2   D : | 42 |  |  |  |  |  |

Table I.33: Heuristic solution of the 4M3A.s instance.



| c | i | : | p | s | m | i | : | p | s | m | i  | : | p | s | m | i  |
|---|---|---|---|---|---|---|---|---|---|---|----|---|---|---|---|----|
| 1 |   | A | : | 1 | 0 | 1 |   | 1 | : | 2 | 42 | 1 |   | D | : | 45 |
| 2 |   | A | : | 3 | 0 | 1 |   | 1 | : | 4 | 28 | 1 |   | D | : | 33 |
| 3 |   | A | : | 5 | 0 | 1 |   | 1 | : | 6 | 20 | 1 |   | D | : | 27 |
| 4 |   | A | : | 7 | 0 | 2 |   | 2 | : | 9 | 28 | 2 |   | D | : | 39 |

Table I.34: Heuristic solution of the 4M3B.s instance.

| c | i | : | p | s | m | i | : | p | s | m  | i  | : | p | s | m | i  |
|---|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|----|
| 1 |   | A | : | 1 | 0 | 1 |   | 1 | : | 2  | 33 | 1 |   | D | : | 36 |
| 2 |   | A | : | 3 | 0 | 2 |   | 2 | : | 5  | 41 | 2 |   | D | : | 48 |
| 3 |   | A | : | 5 | 2 | 2 |   | 2 | : | 9  | 31 | 2 |   | D | : | 42 |
| 4 |   | A | : | 7 | 4 | 1 |   | 1 | : | 12 | 17 | 1 |   | D | : | 30 |

Table I.35: Heuristic solution of the 4M4A.s instance.

| c | i | : | p | s | m | i | : | p | s | m  | i  | : | p | s | m | i  |
|---|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|----|
| 1 |   | A | : | 1 | 0 | 1 |   | 1 | : | 2  | 51 | 1 |   | D | : | 54 |
| 2 |   | A | : | 3 | 0 | 1 |   | 1 | : | 4  | 37 | 1 |   | D | : | 42 |
| 3 |   | A | : | 5 | 0 | 2 |   | 2 | : | 7  | 39 | 2 |   | D | : | 48 |
| 4 |   | A | : | 7 | 2 | 1 |   | 1 | : | 10 | 25 | 1 |   | D | : | 36 |

Table I.36: Heuristic solution of the 4M4B.s instance.

| c   i : |     |     | p s m   i : |      |         | p s m   i : |   |   | p s m   i : |    |  | p s m   i : |  |  |
|---------|-----|-----|-------------|------|---------|-------------|---|---|-------------|----|--|-------------|--|--|
| 1       | A : | 1 0 | 1   2 :     | 2 11 | 1   D : | 14          |   |   |             |    |  |             |  |  |
| 2       | A : | 3 0 | 1   2 :     | 4 8  | 1   3 : | 13          | 1 | 2 | D :         | 16 |  |             |  |  |
| 3       | A : | 5 0 | 1   2 :     | 6 5  | 1   1 : | 12          | 4 | 2 | D :         | 18 |  |             |  |  |
| 4       | A : | 7 0 | 2   1 :     | 9 9  | 2   D : | 20          |   |   |             |    |  |             |  |  |
| 5       | A : | 9 0 | 2   3 :     | 11 9 | 2   D : | 22          |   |   |             |    |  |             |  |  |

Table I.37: Heuristic solution of the 5S.t instance.

| c | i | : | p | s | m | i | : | p | s | m  | i  | : | p | s | m | i  |
|---|---|---|---|---|---|---|---|---|---|----|----|---|---|---|---|----|
| 1 |   | A | : | 1 | 0 | 1 |   | 2 | : | 2  | 19 | 1 |   | D | : | 22 |
| 2 |   | A | : | 3 | 0 | 1 |   | 2 | : | 4  | 15 | 1 |   | D | : | 20 |
| 3 |   | A | : | 5 | 0 | 1 |   | 2 | : | 6  | 11 | 1 |   | D | : | 18 |
| 4 |   | A | : | 7 | 0 | 2 |   | 1 | : | 9  | 5  | 2 |   | D | : | 16 |
| 5 |   | A | : | 9 | 0 | 2 |   | 1 | : | 11 | 1  | 2 |   | D | : | 14 |

Table I.38: Heuristic solution of the 5C.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 9  | 1 | D : | 12 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 11 | 2 | D : | 18 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 5  | 2 | D : | 14 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 3 : | 9  | 9  | 2 | D : | 20 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 3 : | 11 | 3  | 2 | D : | 16 |  |  |

Table I.39: Heuristic solution of the 5M1A.τ instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 9  | 1 | D : | 12 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 9  | 2 | D : | 16 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 3 : | 7  | 11 | 2 | D : | 20 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 3  | 2 | D : | 14 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 3 : | 11 | 5  | 2 | D : | 18 |  |  |

Table I.40: Heuristic solution of the 5M1B.τ instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 13 | 1 | D : | 16 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 7  | 1 | D : | 12 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 9  | 2 | D : | 18 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 3  | 2 | D : | 14 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 3 : | 11 | 7  | 2 | D : | 20 |  |  |

Table I.41: Heuristic solution of the 5M2A.τ instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 15 | 1 | D : | 18 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 7  | 1 | D : | 12 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 7  | 2 | D : | 16 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 3 : | 9  | 9  | 2 | D : | 20 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 1 : | 11 | 1  | 2 | D : | 14 |  |  |

Table I.42: Heuristic solution of the 5M2B.τ instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 11 | 1 | D : | 14 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 11 | 2 | D : | 18 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 2 : | 6  | 5  | 1 | D : | 12 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 5  | 2 | D : | 16 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 3 : | 11 | 7  | 2 | D : | 20 |  |  |

Table I.43: Heuristic solution of the 5M3A.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 13 | 1 | D : | 16 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 13 | 2 | D : | 20 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 2 : | 6  | 5  | 1 | D : | 12 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 7  | 2 | D : | 18 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 1 : | 11 | 1  | 2 | D : | 14 |  |  |

Table I.44: Heuristic solution of the 5M3B.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 11 | 1 | D : | 14 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 13 | 2 | D : | 20 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 7  | 2 | D : | 16 |  |  |
| 4                                     | A : | 7 | 0 | 1 | 2 : | 8  | 3  | 1 | D : | 12 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 3 : | 11 | 5  | 2 | D : | 18 |  |  |

Table I.45: Heuristic solution of the 5M4A.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 13 | 1 | D : | 16 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 9  | 1 | D : | 14 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 9  | 2 | D : | 18 |  |  |
| 4                                     | A : | 7 | 0 | 1 | 2 : | 8  | 3  | 1 | D : | 12 |  |  |
| 5                                     | A : | 9 | 0 | 2 | 3 : | 11 | 7  | 2 | D : | 20 |  |  |

Table I.46: Heuristic solution of the 5M4B.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 15 | 1 | D : | 18 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 9  | 1 | D : | 14 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 11 | 2 | D : | 20 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 5  | 2 | D : | 16 |  |  |
| 5                                     | A : | 9 | 0 | 1 | 2 : | 10 | 1  | 1 | D : | 12 |  |  |

Table I.47: Heuristic solution of the 5M5A.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 13 | 1 | D : | 16 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 13 | 2 | D : | 20 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 2 : | 6  | 7  | 1 | D : | 14 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 7  | 2 | D : | 18 |  |  |
| 5                                     | A : | 9 | 0 | 1 | 2 : | 10 | 1  | 1 | D : | 12 |  |  |

Table I.48: Heuristic solution of the 5M5B.t instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |    |            |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|----|------------|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 39 | 1 | D : | 42 |    |            |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 12 | 1 | 3 : | 17 | 29 | 2   D : 48 |
| 3                                     | A : | 5 | 0 | 1 | 2 : | 6  | 8  | 1 | 1 : | 15 | 37 | 2   D : 54 |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 49 | 2 | D : | 60 |    |            |
| 5                                     | A : | 9 | 2 | 2 | 3 : | 13 | 51 | 2 | D : | 66 |    |            |

Table I.49: Heuristic solution of the 5S.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 63 | 1 | D : | 66 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 55 | 1 | D : | 60 |  |  |
| 3                                     | A : | 5 | 0 | 1 | 2 : | 6  | 47 | 1 | D : | 54 |  |  |
| 4                                     | A : | 7 | 0 | 2 | 1 : | 9  | 37 | 2 | D : | 48 |  |  |
| 5                                     | A : | 9 | 2 | 2 | 1 : | 13 | 27 | 2 | D : | 42 |  |  |

Table I.50: Heuristic solution of the 5C.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 33 | 1 | D : | 36 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 47 | 2 | D : | 54 |  |  |
| 3                                     | A : | 5 | 2 | 2 | 1 : | 9  | 31 | 2 | D : | 42 |  |  |
| 4                                     | A : | 7 | 4 | 2 | 3 : | 13 | 45 | 2 | D : | 60 |  |  |
| 5                                     | A : | 9 | 6 | 2 | 3 : | 17 | 29 | 2 | D : | 48 |  |  |

Table I.51: Heuristic solution of the 5M1A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 33 | 1 | D : | 36 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 41 | 2 | D : | 48 |  |  |
| 3                                     | A : | 5 | 2 | 2 | 3 : | 9  | 49 | 2 | D : | 60 |  |  |
| 4                                     | A : | 7 | 4 | 2 | 1 : | 13 | 27 | 2 | D : | 42 |  |  |
| 5                                     | A : | 9 | 6 | 2 | 3 : | 17 | 35 | 2 | D : | 54 |  |  |

Table I.52: Heuristic solution of the 5M1B.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 45 | 1 | D : | 48 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 31 | 1 | D : | 36 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 45 | 2 | D : | 54 |  |  |
| 4                                     | A : | 7 | 2 | 2 | 1 : | 11 | 29 | 2 | D : | 42 |  |  |
| 5                                     | A : | 9 | 4 | 2 | 3 : | 15 | 43 | 2 | D : | 60 |  |  |

Table I.53: Heuristic solution of the 5M2A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 51 | 1 | D : | 54 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 31 | 1 | D : | 36 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 39 | 2 | D : | 48 |  |  |
| 4                                     | A : | 7 | 2 | 2 | 3 : | 11 | 47 | 2 | D : | 60 |  |  |
| 5                                     | A : | 9 | 4 | 2 | 1 : | 15 | 25 | 2 | D : | 42 |  |  |

Table I.54: Heuristic solution of the 5M2B.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 39 | 1 | D : | 42 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 47 | 2 | D : | 54 |  |  |
| 3                                     | A : | 5 | 2 | 1 | 2 : | 8  | 27 | 1 | D : | 36 |  |  |
| 4                                     | A : | 7 | 2 | 2 | 1 : | 11 | 35 | 2 | D : | 48 |  |  |
| 5                                     | A : | 9 | 4 | 2 | 3 : | 15 | 43 | 2 | D : | 60 |  |  |

Table I.55: Heuristic solution of the 5M3A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 45 | 1 | D : | 48 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 53 | 2 | D : | 60 |  |  |
| 3                                     | A : | 5 | 2 | 1 | 2 : | 8  | 27 | 1 | D : | 36 |  |  |
| 4                                     | A : | 7 | 2 | 2 | 1 : | 11 | 41 | 2 | D : | 54 |  |  |
| 5                                     | A : | 9 | 4 | 2 | 1 : | 15 | 25 | 2 | D : | 42 |  |  |

Table I.56: Heuristic solution of the 5M3B.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 39 | 1 | D : | 42 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 53 | 2 | D : | 60 |  |  |
| 3                                     | A : | 5 | 2 | 2 | 1 : | 9  | 37 | 2 | D : | 48 |  |  |
| 4                                     | A : | 7 | 4 | 1 | 2 : | 12 | 23 | 1 | D : | 36 |  |  |
| 5                                     | A : | 9 | 4 | 2 | 3 : | 15 | 37 | 2 | D : | 54 |  |  |

Table I.57: Heuristic solution of the 5M4A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 45 | 1 | D : | 48 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 37 | 1 | D : | 42 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 45 | 2 | D : | 54 |  |  |
| 4                                     | A : | 7 | 2 | 1 | 2 : | 10 | 25 | 1 | D : | 36 |  |  |
| 5                                     | A : | 9 | 2 | 2 | 3 : | 13 | 45 | 2 | D : | 60 |  |  |

Table I.58: Heuristic solution of the 5M4B.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 51 | 1 | D : | 54 |  |  |
| 2                                     | A : | 3 | 0 | 1 | 2 : | 4  | 37 | 1 | D : | 42 |  |  |
| 3                                     | A : | 5 | 0 | 2 | 1 : | 7  | 51 | 2 | D : | 60 |  |  |
| 4                                     | A : | 7 | 2 | 2 | 1 : | 11 | 35 | 2 | D : | 48 |  |  |
| 5                                     | A : | 9 | 4 | 1 | 2 : | 14 | 21 | 1 | D : | 36 |  |  |

Table I.59: Heuristic solution of the 5M5A.s instance.

| c   i : p s m   i : p s m   i : p s m |     |   |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|---|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1 | 0 | 1 | 2 : | 2  | 45 | 1 | D : | 48 |  |  |
| 2                                     | A : | 3 | 0 | 2 | 1 : | 5  | 53 | 2 | D : | 60 |  |  |
| 3                                     | A : | 5 | 2 | 1 | 2 : | 8  | 33 | 1 | D : | 42 |  |  |
| 4                                     | A : | 7 | 2 | 2 | 1 : | 11 | 41 | 2 | D : | 54 |  |  |
| 5                                     | A : | 9 | 4 | 1 | 2 : | 14 | 21 | 1 | D : | 36 |  |  |

Table I.60: Heuristic solution of the 5M5B.s instance.

## I.2 Medium-scale class instances

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |   |   |     |    |  |  |
|---------------------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|---|---|-----|----|--|--|
| 1                                                 | A : | 1  | 0 | 1 | 2 : | 2  | 15 | 1 | D : | 18 |   |   |     |    |  |  |
| 2                                                 | A : | 3  | 0 | 1 | 2 : | 4  | 10 | 1 | 1 : | 15 | 3 | 2 | D : | 20 |  |  |
| 3                                                 | A : | 5  | 0 | 1 | 2 : | 6  | 7  | 1 | 1 : | 14 | 6 | 2 | D : | 22 |  |  |
| 4                                                 | A : | 7  | 0 | 2 | 1 : | 9  | 13 | 2 | D : | 24 |   |   |     |    |  |  |
| 5                                                 | A : | 9  | 0 | 2 | 3 : | 11 | 14 | 2 | D : | 27 |   |   |     |    |  |  |
| 6                                                 | A : | 11 | 0 | 2 | 3 : | 13 | 11 | 1 | 2 : | 25 | 2 | 1 | D : | 28 |  |  |

Table I.61: Heuristic solution of the 6S.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1  | 0 | 1 | 2 : | 2  | 23 | 1 | D : | 26 |  |  |
| 2                                     | A : | 3  | 0 | 1 | 2 : | 4  | 19 | 1 | D : | 24 |  |  |
| 3                                     | A : | 5  | 0 | 1 | 2 : | 6  | 15 | 1 | D : | 22 |  |  |
| 4                                     | A : | 7  | 0 | 2 | 1 : | 9  | 9  | 2 | D : | 20 |  |  |
| 5                                     | A : | 9  | 0 | 2 | 1 : | 11 | 5  | 2 | D : | 18 |  |  |
| 6                                     | A : | 11 | 0 | 2 | 1 : | 13 | 1  | 2 | D : | 16 |  |  |

Table I.62: Heuristic solution of the 6C.t instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m   i : | p   | s  | m |
|---------|----|---|---------|----|----|---------|----|---|---------|-----|----|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 11 | 1   D : | 14 |   |         |     |    |   |
| 2   A : | 3  | 0 | 2   1 : | 5  | 11 | 2   D : | 18 |   |         |     |    |   |
| 3   A : | 5  | 0 | 2   3 : | 7  | 14 | 2   D : | 23 |   |         |     |    |   |
| 4   A : | 7  | 0 | 2   1 : | 9  | 5  | 2   D : | 16 |   |         |     |    |   |
| 5   A : | 9  | 0 | 2   3 : | 11 | 8  | 2   D : | 21 |   |         |     |    |   |
| 6   A : | 11 | 0 | 2   3 : | 13 | 5  | 1   2 : | 19 | 4 | 1       | D : | 24 |   |

Table I.63: Heuristic solution of the 6M1.t instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 18 | 1   D : | 21 |   |   |
| 2   A : | 3  | 0 | 1   2 : | 4  | 10 | 1   D : | 15 |   |   |
| 3   A : | 5  | 0 | 2   1 : | 7  | 14 | 2   D : | 23 |   |   |
| 4   A : | 7  | 0 | 2   1 : | 9  | 6  | 2   D : | 17 |   |   |
| 5   A : | 9  | 0 | 2   3 : | 11 | 12 | 2   D : | 25 |   |   |
| 6   A : | 11 | 0 | 2   3 : | 13 | 4  | 2   D : | 19 |   |   |

Table I.64: Heuristic solution of the 6M2.t instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 18 | 1   D : | 21 |   |   |
| 2   A : | 3  | 0 | 1   2 : | 4  | 12 | 1   D : | 17 |   |   |
| 3   A : | 5  | 0 | 1   2 : | 6  | 7  | 1   D : | 14 |   |   |
| 4   A : | 7  | 0 | 2   1 : | 9  | 12 | 2   D : | 23 |   |   |
| 5   A : | 9  | 0 | 2   1 : | 11 | 6  | 2   D : | 19 |   |   |
| 6   A : | 11 | 0 | 2   1 : | 13 | 1  | 2   D : | 16 |   |   |

Table I.65: Heuristic solution of the 6M3.t instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 21 | 1   D : | 24 |   |   |
| 2   A : | 3  | 0 | 1   2 : | 4  | 11 | 1   D : | 16 |   |   |
| 3   A : | 5  | 0 | 2   1 : | 7  | 11 | 2   D : | 20 |   |   |
| 4   A : | 7  | 0 | 1   2 : | 8  | 5  | 1   D : | 14 |   |   |
| 5   A : | 9  | 0 | 2   1 : | 11 | 5  | 2   D : | 18 |   |   |
| 6   A : | 11 | 0 | 2   3 : | 13 | 7  | 2   D : | 22 |   |   |

Table I.66: Heuristic solution of the 6M4.t instance.



| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 17 | 1 | D | : | 20 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 11 | 1 | D | : | 16 |   |   |   |
| 3 | A | : | 5  | 0 | 2 | 1 | : | 7  | 13 | 2 | D | : | 22 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 1 | : | 9  | 7  | 2 | D | : | 18 |   |   |   |
| 5 | A | : | 9  | 0 | 1 | 2 | : | 10 | 3  | 1 | D | : | 14 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 3 | : | 13 | 9  | 2 | D | : | 24 |   |   |   |

Table I.67: Heuristic solution of the **6M5.t** instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 21 | 1 | D | : | 24 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 15 | 1 | D | : | 20 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 2 | : | 6  | 9  | 1 | D | : | 16 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 1 | : | 9  | 11 | 2 | D | : | 22 |   |   |   |
| 5 | A | : | 9  | 0 | 2 | 1 | : | 11 | 5  | 2 | D | : | 18 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 1 | : | 13 | 0  | 2 | D | : | 15 |   |   |   |

Table I.68: Heuristic solution of the **6M6.t** instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s  | m | i      |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|----|---|--------|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 51 | 1 | D | : | 54 |    |   |        |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 16 | 1 | 1 | : | 21 | 37 | 2 | D : 60 |
| 3 | A | : | 5  | 0 | 1 | 2 | : | 6  | 12 | 1 | 1 | : | 19 | 45 | 2 | D : 66 |
| 4 | A | : | 7  | 0 | 2 | 1 | : | 9  | 61 | 2 | D | : | 72 |    |   |        |
| 5 | A | : | 9  | 2 | 2 | 3 | : | 13 | 63 | 2 | D | : | 78 |    |   |        |
| 6 | A | : | 11 | 4 | 2 | 3 | : | 17 | 57 | 1 | 2 | : | 75 | 8  | 1 | D : 84 |

Table I.69: Heuristic solution of the **6S.s** instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 75 | 1 | D | : | 78 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 67 | 1 | D | : | 72 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 2 | : | 6  | 59 | 1 | D | : | 66 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 1 | : | 9  | 49 | 2 | D | : | 60 |   |   |   |
| 5 | A | : | 9  | 2 | 2 | 1 | : | 13 | 39 | 2 | D | : | 54 |   |   |   |
| 6 | A | : | 11 | 4 | 2 | 1 | : | 17 | 29 | 2 | D | : | 48 |   |   |   |

Table I.70: Heuristic solution of the **6C.s** instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 39 | 1   D : | 42 |    |         |    |   |   |
| 2   A : | 3  | 0 | 2   1 : | 5  | 47 | 2   D : | 54 |    |         |    |   |   |
| 3   A : | 5  | 2 | 2   3 : | 9  | 55 | 2   D : | 66 |    |         |    |   |   |
| 4   A : | 7  | 4 | 2   1 : | 13 | 33 | 2   D : | 48 |    |         |    |   |   |
| 5   A : | 9  | 6 | 2   3 : | 17 | 41 | 2   D : | 60 |    |         |    |   |   |
| 6   A : | 11 | 8 | 2   3 : | 21 | 35 | 1   2 : | 57 | 14 | 1   D : | 72 |   |   |

Table I.71: Heuristic solution of the 6M1.s instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 60 | 1   D : | 63 |   |   |
| 2   A : | 3  | 0 | 1   2 : | 4  | 40 | 1   D : | 45 |   |   |
| 3   A : | 5  | 0 | 2   1 : | 7  | 60 | 2   D : | 69 |   |   |
| 4   A : | 7  | 2 | 2   1 : | 11 | 38 | 2   D : | 51 |   |   |
| 5   A : | 9  | 4 | 2   3 : | 15 | 58 | 2   D : | 75 |   |   |
| 6   A : | 11 | 6 | 2   3 : | 19 | 36 | 2   D : | 57 |   |   |

Table I.72: Heuristic solution of the 6M2.s instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 60 | 1   D : | 63 |   |   |
| 2   A : | 3  | 0 | 1   2 : | 4  | 46 | 1   D : | 51 |   |   |
| 3   A : | 5  | 0 | 1   2 : | 6  | 32 | 1   D : | 39 |   |   |
| 4   A : | 7  | 0 | 2   1 : | 9  | 58 | 2   D : | 69 |   |   |
| 5   A : | 9  | 2 | 2   1 : | 13 | 42 | 2   D : | 57 |   |   |
| 6   A : | 11 | 4 | 2   1 : | 17 | 26 | 2   D : | 45 |   |   |

Table I.73: Heuristic solution of the 6M3.s instance.

| c   i : | p  | s | m   i : | p  | s  | m   i : | p  | s | m |
|---------|----|---|---------|----|----|---------|----|---|---|
| 1   A : | 1  | 0 | 1   2 : | 2  | 69 | 1   D : | 72 |   |   |
| 2   A : | 3  | 0 | 1   2 : | 4  | 43 | 1   D : | 48 |   |   |
| 3   A : | 5  | 0 | 2   1 : | 7  | 51 | 2   D : | 60 |   |   |
| 4   A : | 7  | 2 | 1   2 : | 10 | 31 | 1   D : | 42 |   |   |
| 5   A : | 9  | 2 | 2   1 : | 13 | 39 | 2   D : | 54 |   |   |
| 6   A : | 11 | 4 | 2   3 : | 17 | 47 | 2   D : | 66 |   |   |

Table I.74: Heuristic solution of the 6M4.s instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 57 | 1 | D | : | 60 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 43 | 1 | D | : | 48 |   |   |   |
| 3 | A | : | 5  | 0 | 2 | 1 | : | 7  | 57 | 2 | D | : | 66 |   |   |   |
| 4 | A | : | 7  | 2 | 2 | 1 | : | 11 | 41 | 2 | D | : | 54 |   |   |   |
| 5 | A | : | 9  | 4 | 1 | 2 | : | 14 | 27 | 1 | D | : | 42 |   |   |   |
| 6 | A | : | 11 | 4 | 2 | 3 | : | 17 | 53 | 2 | D | : | 72 |   |   |   |

Table I.75: Heuristic solution of the 6M5.s instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 69 | 1 | D | : | 72 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 55 | 1 | D | : | 60 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 2 | : | 6  | 41 | 1 | D | : | 48 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 1 | : | 9  | 55 | 2 | D | : | 66 |   |   |   |
| 5 | A | : | 9  | 2 | 2 | 1 | : | 13 | 39 | 2 | D | : | 54 |   |   |   |
| 6 | A | : | 11 | 4 | 2 | 1 | : | 17 | 23 | 2 | D | : | 42 |   |   |   |

Table I.76: Heuristic solution of the 6M6.s instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 2 | : | 2  | 75 | 1 | D | : | 78 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 2 | : | 4  | 67 | 1 | D | : | 72 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 2 | : | 6  | 59 | 1 | D | : | 66 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 1 | : | 9  | 49 | 2 | D | : | 60 |   |   |   |
| 5 | A | : | 9  | 2 | 2 | 1 | : | 13 | 39 | 2 | D | : | 54 |   |   |   |
| 6 | A | : | 11 | 4 | 2 | 1 | : | 17 | 29 | 2 | D | : | 48 |   |   |   |

Table I.77: Heuristic solution of the 8S.t instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 1 | : | 2  | 33 | 2 | D | : | 37 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 1 | : | 4  | 29 | 2 | D | : | 35 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 1 | : | 6  | 25 | 2 | D | : | 33 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 2 | : | 9  | 19 | 3 | D | : | 31 |   |   |   |
| 5 | A | : | 9  | 0 | 2 | 2 | : | 11 | 14 | 3 | D | : | 28 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 2 | : | 13 | 9  | 3 | D | : | 25 |   |   |   |
| 7 | A | : | 13 | 0 | 2 | 3 | : | 15 | 6  | 1 | D | : | 22 |   |   |   |
| 8 | A | : | 15 | 0 | 2 | 3 | : | 17 | 2  | 1 | D | : | 20 |   |   |   |

Table I.78: Heuristic solution of the 8C.t instance.

| c   i : p s m   i : p s m   i : p s m |    |   |         |    |    |         |    |  |  |  |  |
|---------------------------------------|----|---|---------|----|----|---------|----|--|--|--|--|
| -----                                 |    |   |         |    |    |         |    |  |  |  |  |
| 1   A :                               | 1  | 0 | 1   1 : | 2  | 18 | 2   D : | 22 |  |  |  |  |
| 2   A :                               | 3  | 0 | 2   2 : | 5  | 20 | 3   D : | 28 |  |  |  |  |
| 3   A :                               | 5  | 0 | 2   3 : | 7  | 26 | 1   D : | 34 |  |  |  |  |
| 4   A :                               | 7  | 0 | 2   2 : | 9  | 13 | 3   D : | 25 |  |  |  |  |
| 5   A :                               | 9  | 0 | 2   3 : | 11 | 18 | 1   D : | 30 |  |  |  |  |
| 6   A :                               | 11 | 0 | 3   4 : | 14 | 20 | 2   D : | 36 |  |  |  |  |
| 7   A :                               | 13 | 1 | 2   3 : | 16 | 12 | 1   D : | 29 |  |  |  |  |
| 8   A :                               | 15 | 1 | 3   4 : | 19 | 11 | 2   D : | 32 |  |  |  |  |

Table I.79: Heuristic solution of the 8M1.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|
| -----                                 |     |    |   |   |     |    |    |   |     |    |  |
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 26 | 2 | D : | 30 |  |
| 2                                     | A : | 3  | 0 | 1 | 1 : | 4  | 16 | 2 | D : | 22 |  |
| 3                                     | A : | 5  | 0 | 2 | 2 : | 7  | 23 | 3 | D : | 33 |  |
| 4                                     | A : | 7  | 0 | 2 | 2 : | 9  | 13 | 3 | D : | 25 |  |
| 5                                     | A : | 9  | 0 | 2 | 3 : | 11 | 22 | 1 | D : | 34 |  |
| 6                                     | A : | 11 | 0 | 2 | 3 : | 13 | 12 | 1 | D : | 26 |  |
| 7                                     | A : | 13 | 0 | 3 | 4 : | 16 | 18 | 2 | D : | 36 |  |
| 8                                     | A : | 15 | 1 | 3 | 4 : | 19 | 7  | 2 | D : | 28 |  |

Table I.80: Heuristic solution of the 8M2.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|
| -----                                 |     |    |   |   |     |    |    |   |     |    |  |
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 19 | 2 | D : | 23 |  |
| 2                                     | A : | 3  | 0 | 2 | 2 : | 5  | 26 | 3 | D : | 34 |  |
| 3                                     | A : | 5  | 0 | 1 | 1 : | 6  | 13 | 2 | D : | 21 |  |
| 4                                     | A : | 7  | 0 | 2 | 2 : | 9  | 19 | 3 | D : | 31 |  |
| 5                                     | A : | 9  | 0 | 2 | 2 : | 11 | 13 | 3 | D : | 27 |  |
| 6                                     | A : | 11 | 0 | 2 | 3 : | 13 | 23 | 1 | D : | 37 |  |
| 7                                     | A : | 13 | 0 | 2 | 3 : | 15 | 12 | 1 | D : | 28 |  |
| 8                                     | A : | 15 | 0 | 3 | 4 : | 18 | 16 | 2 | D : | 36 |  |

Table I.81: Heuristic solution of the 8M3.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|
| -----                                 |     |    |   |   |     |    |    |   |     |    |  |
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 29 | 2 | D : | 33 |  |
| 2                                     | A : | 3  | 0 | 1 | 1 : | 4  | 17 | 2 | D : | 23 |  |
| 3                                     | A : | 5  | 0 | 2 | 2 : | 7  | 19 | 3 | D : | 29 |  |
| 4                                     | A : | 7  | 0 | 1 | 1 : | 8  | 10 | 2 | D : | 20 |  |
| 5                                     | A : | 9  | 0 | 2 | 2 : | 11 | 12 | 3 | D : | 26 |  |
| 6                                     | A : | 11 | 0 | 2 | 3 : | 13 | 20 | 1 | D : | 34 |  |
| 7                                     | A : | 13 | 0 | 2 | 3 : | 15 | 5  | 1 | D : | 21 |  |
| 8                                     | A : | 15 | 0 | 3 | 4 : | 18 | 11 | 2 | D : | 31 |  |

Table I.82: Heuristic solution of the 8M4.t instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 1 | : | 2  | 30 | 2 | D | : | 34 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 1 | : | 4  | 24 | 2 | D | : | 30 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 1 | : | 6  | 15 | 2 | D | : | 23 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 2 | : | 9  | 16 | 3 | D | : | 28 |   |   |   |
| 5 | A | : | 9  | 0 | 2 | 2 | : | 11 | 7  | 3 | D | : | 21 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 3 | : | 13 | 11 | 1 | D | : | 25 |   |   |   |
| 7 | A | : | 13 | 0 | 3 | 4 | : | 16 | 14 | 2 | D | : | 32 |   |   |   |
| 8 | A | : | 15 | 1 | 2 | 3 | : | 18 | 5  | 1 | D | : | 24 |   |   |   |

Table I.83: Heuristic solution of the 8M5.t instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 1 | : | 2  | 27 | 2 | D | : | 31 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 1 | : | 4  | 23 | 2 | D | : | 29 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 1 | : | 6  | 15 | 2 | D | : | 23 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 2 | : | 9  | 14 | 3 | D | : | 26 |   |   |   |
| 5 | A | : | 9  | 0 | 2 | 3 | : | 11 | 20 | 1 | D | : | 32 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 2 | : | 13 | 5  | 3 | D | : | 21 |   |   |   |
| 7 | A | : | 13 | 0 | 2 | 3 | : | 15 | 11 | 1 | D | : | 27 |   |   |   |
| 8 | A | : | 15 | 0 | 3 | 4 | : | 18 | 14 | 2 | D | : | 34 |   |   |   |

Table I.84: Heuristic solution of the 8M6.t instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 1 | : | 2  | 27 | 2 | D | : | 31 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 1 | : | 4  | 20 | 2 | D | : | 26 |   |   |   |
| 3 | A | : | 5  | 0 | 1 | 1 | : | 6  | 13 | 2 | D | : | 21 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 2 | : | 9  | 22 | 3 | D | : | 34 |   |   |   |
| 5 | A | : | 9  | 0 | 2 | 2 | : | 11 | 15 | 3 | D | : | 29 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 2 | : | 13 | 8  | 3 | D | : | 24 |   |   |   |
| 7 | A | : | 13 | 0 | 2 | 3 | : | 15 | 3  | 1 | D | : | 19 |   |   |   |
| 8 | A | : | 15 | 0 | 3 | 4 | : | 18 | 16 | 2 | D | : | 36 |   |   |   |

Table I.85: Heuristic solution of the 8M7.t instance.

| c | i | : | p  | s | m | i | : | p  | s  | m | i | : | p  | s | m | i |
|---|---|---|----|---|---|---|---|----|----|---|---|---|----|---|---|---|
| 1 | A | : | 1  | 0 | 1 | 1 | : | 2  | 26 | 2 | D | : | 30 |   |   |   |
| 2 | A | : | 3  | 0 | 1 | 1 | : | 4  | 17 | 2 | D | : | 23 |   |   |   |
| 3 | A | : | 5  | 0 | 2 | 2 | : | 7  | 23 | 3 | D | : | 33 |   |   |   |
| 4 | A | : | 7  | 0 | 2 | 2 | : | 9  | 16 | 3 | D | : | 28 |   |   |   |
| 5 | A | : | 9  | 0 | 1 | 1 | : | 10 | 9  | 2 | D | : | 21 |   |   |   |
| 6 | A | : | 11 | 0 | 2 | 3 | : | 13 | 20 | 1 | D | : | 34 |   |   |   |
| 7 | A | : | 13 | 0 | 2 | 3 | : | 15 | 9  | 1 | D | : | 25 |   |   |   |
| 8 | A | : | 15 | 0 | 2 | 3 | : | 17 | 1  | 1 | D | : | 19 |   |   |   |

Table I.86: Heuristic solution of the 8M8.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |    |   |         |    |    |         |     |    |         |     |  |  |
|---------------------------------------------------|----|---|---------|----|----|---------|-----|----|---------|-----|--|--|
| 1   A :                                           | 1  | 0 | 1   1 : | 2  | 71 | 2   D : | 75  |    |         |     |  |  |
| 2   A :                                           | 3  | 0 | 1   1 : | 4  | 26 | 1   3 : | 31  | 49 | 1   D : | 81  |  |  |
| 3   A :                                           | 5  | 0 | 1   1 : | 6  | 22 | 1   3 : | 29  | 65 | 1   D : | 95  |  |  |
| 4   A :                                           | 7  | 0 | 2   2 : | 9  | 81 | 3   D : | 93  |    |         |     |  |  |
| 5   A :                                           | 9  | 2 | 2   2 : | 13 | 73 | 1   4 : | 87  | 10 | 2   D : | 99  |  |  |
| 6   A :                                           | 11 | 4 | 2   2 : | 17 | 67 | 1   4 : | 85  | 18 | 2   D : | 105 |  |  |
| 7   A :                                           | 13 | 6 | 2   3 : | 21 | 89 | 1   D : | 111 |    |         |     |  |  |
| 8   A :                                           | 15 | 8 | 3   4 : | 26 | 89 | 2   D : | 117 |    |         |     |  |  |

Table I.87: Heuristic solution of the 8S.s instance.

| c   i : p s m   i : p s m   i : p s m |    |   |         |    |    |         |     |  |  |  |  |  |
|---------------------------------------|----|---|---------|----|----|---------|-----|--|--|--|--|--|
| 1   A :                               | 1  | 0 | 1   1 : | 2  | 98 | 2   D : | 102 |  |  |  |  |  |
| 2   A :                               | 3  | 0 | 1   1 : | 4  | 90 | 2   D : | 96  |  |  |  |  |  |
| 3   A :                               | 5  | 0 | 1   1 : | 6  | 82 | 2   D : | 90  |  |  |  |  |  |
| 4   A :                               | 7  | 0 | 2   2 : | 9  | 72 | 3   D : | 84  |  |  |  |  |  |
| 5   A :                               | 9  | 2 | 2   2 : | 13 | 62 | 3   D : | 78  |  |  |  |  |  |
| 6   A :                               | 11 | 4 | 2   2 : | 17 | 52 | 3   D : | 72  |  |  |  |  |  |
| 7   A :                               | 13 | 6 | 2   3 : | 21 | 44 | 1   D : | 66  |  |  |  |  |  |
| 8   A :                               | 15 | 8 | 2   3 : | 25 | 34 | 1   D : | 60  |  |  |  |  |  |

Table I.88: Heuristic solution of the 8C.s instance.

| c   i : p s m   i : p s m   i : p s m |    |    |         |    |    |         |     |  |  |  |  |  |
|---------------------------------------|----|----|---------|----|----|---------|-----|--|--|--|--|--|
| 1   A :                               | 1  | 0  | 1   1 : | 2  | 62 | 2   D : | 66  |  |  |  |  |  |
| 2   A :                               | 3  | 0  | 2   2 : | 5  | 76 | 3   D : | 84  |  |  |  |  |  |
| 3   A :                               | 5  | 2  | 2   3 : | 9  | 92 | 1   D : | 102 |  |  |  |  |  |
| 4   A :                               | 7  | 4  | 2   2 : | 13 | 56 | 3   D : | 72  |  |  |  |  |  |
| 5   A :                               | 9  | 6  | 2   3 : | 17 | 72 | 1   D : | 90  |  |  |  |  |  |
| 6   A :                               | 11 | 8  | 3   4 : | 22 | 84 | 2   D : | 108 |  |  |  |  |  |
| 7   A :                               | 13 | 12 | 2   3 : | 27 | 50 | 1   D : | 78  |  |  |  |  |  |
| 8   A :                               | 15 | 14 | 3   4 : | 32 | 62 | 2   D : | 96  |  |  |  |  |  |

Table I.89: Heuristic solution of the 8M1.s instance.

| c   i : p s m   i : p s m   i : p s m |    |    |         |    |    |         |     |  |  |  |  |  |
|---------------------------------------|----|----|---------|----|----|---------|-----|--|--|--|--|--|
| 1   A :                               | 1  | 0  | 1   1 : | 2  | 86 | 2   D : | 90  |  |  |  |  |  |
| 2   A :                               | 3  | 0  | 1   1 : | 4  | 60 | 2   D : | 66  |  |  |  |  |  |
| 3   A :                               | 5  | 0  | 2   2 : | 7  | 86 | 3   D : | 96  |  |  |  |  |  |
| 4   A :                               | 7  | 2  | 2   2 : | 11 | 58 | 3   D : | 72  |  |  |  |  |  |
| 5   A :                               | 9  | 4  | 2   3 : | 15 | 86 | 1   D : | 102 |  |  |  |  |  |
| 6   A :                               | 11 | 6  | 2   3 : | 19 | 58 | 1   D : | 78  |  |  |  |  |  |
| 7   A :                               | 13 | 8  | 3   4 : | 24 | 82 | 2   D : | 108 |  |  |  |  |  |
| 8   A :                               | 15 | 12 | 3   4 : | 30 | 52 | 2   D : | 84  |  |  |  |  |  |

Table I.90: Heuristic solution of the 8M2.s instance.

| c   i : p s m   i : p s m   i : p s m |     |    |    |   |     |    |    |   |     |     |  |
|---------------------------------------|-----|----|----|---|-----|----|----|---|-----|-----|--|
| -----                                 |     |    |    |   |     |    |    |   |     |     |  |
| 1                                     | A : | 1  | 0  | 1 | 1 : | 2  | 65 | 2 | D : | 69  |  |
| 2                                     | A : | 3  | 0  | 2 | 2 : | 5  | 85 | 3 | D : | 93  |  |
| 3                                     | A : | 5  | 2  | 1 | 1 : | 8  | 53 | 2 | D : | 63  |  |
| 4                                     | A : | 7  | 2  | 2 | 2 : | 11 | 73 | 3 | D : | 87  |  |
| 5                                     | A : | 9  | 4  | 2 | 2 : | 15 | 63 | 3 | D : | 81  |  |
| 6                                     | A : | 11 | 6  | 2 | 3 : | 19 | 85 | 1 | D : | 105 |  |
| 7                                     | A : | 13 | 8  | 2 | 3 : | 23 | 51 | 1 | D : | 75  |  |
| 8                                     | A : | 15 | 10 | 3 | 4 : | 28 | 69 | 2 | D : | 99  |  |

Table I.91: Heuristic solution of the 8M3.s instance.

| c   i : p s m   i : p s m   i : p s m |    |   |         |    |    |         |    |  |  |  |  |
|---------------------------------------|----|---|---------|----|----|---------|----|--|--|--|--|
| -----                                 |    |   |         |    |    |         |    |  |  |  |  |
| 1   A :                               | 1  | 0 | 1   1 : | 2  | 86 | 2   D : | 90 |  |  |  |  |
| 2   A :                               | 3  | 0 | 1   1 : | 4  | 60 | 2   D : | 66 |  |  |  |  |
| 3   A :                               | 5  | 0 | 2   2 : | 7  | 68 | 3   D : | 78 |  |  |  |  |
| 4   A :                               | 7  | 2 | 1   1 : | 10 | 42 | 2   D : | 54 |  |  |  |  |
| 5   A :                               | 9  | 2 | 2   2 : | 13 | 56 | 3   D : | 72 |  |  |  |  |
| 6   A :                               | 11 | 4 | 2   3 : | 17 | 78 | 1   D : | 96 |  |  |  |  |
| 7   A :                               | 13 | 6 | 2   3 : | 21 | 38 | 1   D : | 60 |  |  |  |  |
| 8   A :                               | 15 | 8 | 3   4 : | 26 | 56 | 2   D : | 84 |  |  |  |  |

Table I.92: Heuristic solution of the 8M4.s instance.

| c   i : p s m   i : p s m   i : p s m |     |    |    |   |     |    |    |   |     |    |  |
|---------------------------------------|-----|----|----|---|-----|----|----|---|-----|----|--|
| -----                                 |     |    |    |   |     |    |    |   |     |    |  |
| 1                                     | A : | 1  | 0  | 1 | 1 : | 2  | 92 | 2 | D : | 96 |  |
| 2                                     | A : | 3  | 0  | 1 | 1 : | 4  | 78 | 2 | D : | 84 |  |
| 3                                     | A : | 5  | 0  | 1 | 1 : | 6  | 52 | 2 | D : | 60 |  |
| 4                                     | A : | 7  | 0  | 2 | 2 : | 9  | 66 | 3 | D : | 78 |  |
| 5                                     | A : | 9  | 2  | 2 | 2 : | 13 | 38 | 3 | D : | 54 |  |
| 6                                     | A : | 11 | 4  | 2 | 3 : | 17 | 54 | 1 | D : | 72 |  |
| 7                                     | A : | 13 | 6  | 3 | 4 : | 22 | 66 | 2 | D : | 90 |  |
| 8                                     | A : | 15 | 10 | 2 | 3 : | 27 | 38 | 1 | D : | 66 |  |

Table I.93: Heuristic solution of the 8M5.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |    |   |         |    |    |         |    |   |         |    |  |
|---------------------------------------------------|----|---|---------|----|----|---------|----|---|---------|----|--|
| -----                                             |    |   |         |    |    |         |    |   |         |    |  |
| 1   A :                                           | 1  | 0 | 1   1 : | 2  | 77 | 2   D : | 81 |   |         |    |  |
| 2   A :                                           | 3  | 0 | 1   1 : | 4  | 73 | 1   3 : | 78 | 8 | 1   D : | 87 |  |
| 3   A :                                           | 5  | 0 | 1   1 : | 6  | 55 | 2   D : | 63 |   |         |    |  |
| 4   A :                                           | 7  | 0 | 2   2 : | 9  | 63 | 3   D : | 75 |   |         |    |  |
| 5   A :                                           | 9  | 2 | 2   3 : | 13 | 79 | 1   D : | 93 |   |         |    |  |
| 6   A :                                           | 11 | 4 | 2   2 : | 17 | 37 | 3   D : | 57 |   |         |    |  |
| 7   A :                                           | 13 | 6 | 2   3 : | 21 | 47 | 1   D : | 69 |   |         |    |  |
| 8   A :                                           | 15 | 8 | 3   4 : | 26 | 71 | 2   D : | 99 |   |         |    |  |

Table I.94: Heuristic solution of the 8M6.s instance.

|   | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i |
|---|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|---|
| 1 |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 80 | 2 |   | D | : | 84 |   |
| 2 |   | A | : | 3  | 0 | 1 |   | 1 | : | 4  | 66 | 2 |   | D | : | 72 |   |
| 3 |   | A | : | 5  | 0 | 1 |   | 1 | : | 6  | 52 | 2 |   | D | : | 60 |   |
| 4 |   | A | : | 7  | 0 | 2 |   | 2 | : | 9  | 78 | 3 |   | D | : | 90 |   |
| 5 |   | A | : | 9  | 2 | 2 |   | 2 | : | 13 | 62 | 3 |   | D | : | 78 |   |
| 6 |   | A | : | 11 | 4 | 2 |   | 2 | : | 17 | 46 | 3 |   | D | : | 66 |   |
| 7 |   | A | : | 13 | 6 | 2 |   | 3 | : | 21 | 32 | 1 |   | D | : | 54 |   |
| 8 |   | A | : | 15 | 8 | 3 |   | 4 | : | 26 | 68 | 2 |   | D | : | 96 |   |

Table I.95: Heuristic solution of the 8M7.s instance.

|   | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i |
|---|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|---|
| 1 |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 83 | 2 |   | D | : | 87 |   |
| 2 |   | A | : | 3  | 0 | 1 |   | 1 | : | 4  | 63 | 2 |   | D | : | 69 |   |
| 3 |   | A | : | 5  | 0 | 2 |   | 2 | : | 7  | 89 | 3 |   | D | : | 99 |   |
| 4 |   | A | : | 7  | 2 | 2 |   | 2 | : | 11 | 67 | 3 |   | D | : | 81 |   |
| 5 |   | A | : | 9  | 4 | 1 |   | 1 | : | 14 | 47 | 2 |   | D | : | 63 |   |
| 6 |   | A | : | 11 | 4 | 2 |   | 3 | : | 17 | 75 | 1 |   | D | : | 93 |   |
| 7 |   | A | : | 13 | 6 | 2 |   | 3 | : | 21 | 53 | 1 |   | D | : | 75 |   |
| 8 |   | A | : | 15 | 8 | 2 |   | 3 | : | 25 | 31 | 1 |   | D | : | 57 |   |

Table I.96: Heuristic solution of the 8M8.s instance.

|    | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i             |
|----|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|---------------|
| 1  |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 25 | 2 |   | D | : | 29 |               |
| 2  |   | A | : | 5  | 0 | 1 |   | 1 | : | 6  | 19 | 1 |   | 3 | : | 26 | 6 1   D : 33  |
| 3  |   | A | : | 9  | 0 | 1 |   | 1 | : | 10 | 14 | 1 |   | 2 | : | 25 | 10 3   D : 38 |
| 4  |   | A | : | 13 | 0 | 1 |   | 1 | : | 14 | 9  | 1 |   | 2 | : | 24 | 17 3   D : 44 |
| 5  |   | A | : | 17 | 0 | 2 |   | 2 | : | 19 | 28 | 3 |   | D | : | 50 |               |
| 6  |   | A | : | 21 | 0 | 2 |   | 2 | : | 23 | 21 | 3 |   | D | : | 47 |               |
| 7  |   | A | : | 25 | 4 | 1 |   | 1 | : | 30 | 21 | 2 |   | D | : | 53 |               |
| 8  |   | A | : | 29 | 1 | 1 |   | 1 | : | 31 | 19 | 1 |   | 3 | : | 51 | 5 1   D : 57  |
| 9  |   | A | : | 33 | 0 | 2 |   | 3 | : | 35 | 25 | 1 |   | D | : | 61 |               |
| 10 |   | A | : | 37 | 1 | 3 |   | 4 | : | 41 | 22 | 2 |   | D | : | 65 |               |

Table I.97: Heuristic solution of the 10S.t instance.

|    | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i |
|----|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|---|
| 1  |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 74 | 2 |   | D | : | 78 |   |
| 2  |   | A | : | 5  | 0 | 1 |   | 1 | : | 6  | 66 | 2 |   | D | : | 74 |   |
| 3  |   | A | : | 9  | 0 | 1 |   | 1 | : | 10 | 58 | 2 |   | D | : | 70 |   |
| 4  |   | A | : | 13 | 0 | 1 |   | 1 | : | 14 | 50 | 2 |   | D | : | 66 |   |
| 5  |   | A | : | 17 | 0 | 2 |   | 2 | : | 19 | 40 | 3 |   | D | : | 62 |   |
| 6  |   | A | : | 21 | 0 | 2 |   | 2 | : | 23 | 32 | 3 |   | D | : | 58 |   |
| 7  |   | A | : | 25 | 0 | 2 |   | 2 | : | 27 | 24 | 3 |   | D | : | 54 |   |
| 8  |   | A | : | 29 | 0 | 2 |   | 2 | : | 31 | 16 | 3 |   | D | : | 50 |   |
| 9  |   | A | : | 33 | 0 | 2 |   | 3 | : | 35 | 10 | 1 |   | D | : | 46 |   |
| 10 |   | A | : | 37 | 0 | 2 |   | 3 | : | 39 | 2  | 1 |   | D | : | 42 |   |

Table I.98: Heuristic solution of the 10C.t instance.



|    | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i |
|----|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|---|
| 1  |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 25 | 2 |   | D | : | 29 |   |
| 2  |   | A | : | 5  | 0 | 2 |   | 2 | : | 7  | 35 | 3 |   | D | : | 45 |   |
| 3  |   | A | : | 9  | 0 | 2 |   | 3 | : | 11 | 45 | 1 |   | D | : | 57 |   |
| 4  |   | A | : | 13 | 0 | 2 |   | 2 | : | 15 | 17 | 3 |   | D | : | 35 |   |
| 5  |   | A | : | 17 | 0 | 2 |   | 3 | : | 19 | 29 | 1 |   | D | : | 49 |   |
| 6  |   | A | : | 21 | 0 | 3 |   | 4 | : | 24 | 35 | 2 |   | D | : | 61 |   |
| 7  |   | A | : | 25 | 0 | 2 |   | 3 | : | 27 | 9  | 1 |   | D | : | 37 |   |
| 8  |   | A | : | 29 | 0 | 3 |   | 4 | : | 32 | 19 | 2 |   | D | : | 53 |   |
| 9  |   | A | : | 33 | 2 | 1 |   | 1 | : | 36 | 27 | 2 |   | D | : | 65 |   |
| 10 |   | A | : | 37 | 0 | 1 |   | 1 | : | 38 | 1  | 2 |   | D | : | 41 |   |

Table I.99: Heuristic solution of the 10M1.t instance.

|    | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i            |
|----|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|--------------|
| 1  |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 45 | 2 |   | D | : | 49 |              |
| 2  |   | A | : | 5  | 0 | 1 |   | 1 | : | 6  | 22 | 2 |   | D | : | 30 |              |
| 3  |   | A | : | 9  | 0 | 2 |   | 2 | : | 11 | 39 | 3 |   | D | : | 53 |              |
| 4  |   | A | : | 13 | 0 | 2 |   | 2 | : | 15 | 16 | 3 |   | D | : | 34 |              |
| 5  |   | A | : | 17 | 0 | 2 |   | 3 | : | 19 | 37 | 1 |   | D | : | 57 |              |
| 6  |   | A | : | 21 | 0 | 2 |   | 3 | : | 23 | 14 | 1 |   | D | : | 38 |              |
| 7  |   | A | : | 25 | 0 | 3 |   | 4 | : | 28 | 31 | 2 |   | D | : | 61 |              |
| 8  |   | A | : | 29 | 1 | 1 |   | 1 | : | 31 | 9  | 2 |   | D | : | 42 |              |
| 9  |   | A | : | 33 | 1 | 3 |   | 4 | : | 37 | 20 | 1 |   | 2 | : | 58 | 4 3   D : 65 |
| 10 |   | A | : | 37 | 1 | 2 |   | 2 | : | 40 | 2  | 3 |   | D | : | 45 |              |

Table I.100: Heuristic solution of the 10M2.t instance.

|    | c | i | : | p  | s | m | i | : | p | s  | m  | i | : | p | s | m  | i |
|----|---|---|---|----|---|---|---|---|---|----|----|---|---|---|---|----|---|
| 1  |   | A | : | 1  | 0 | 1 |   | 1 | : | 2  | 33 | 2 |   | D | : | 37 |   |
| 2  |   | A | : | 5  | 0 | 2 |   | 2 | : | 7  | 47 | 3 |   | D | : | 57 |   |
| 3  |   | A | : | 9  | 0 | 1 |   | 1 | : | 10 | 17 | 2 |   | D | : | 29 |   |
| 4  |   | A | : | 13 | 0 | 2 |   | 2 | : | 15 | 31 | 3 |   | D | : | 49 |   |
| 5  |   | A | : | 17 | 0 | 2 |   | 2 | : | 19 | 19 | 3 |   | D | : | 41 |   |
| 6  |   | A | : | 21 | 0 | 2 |   | 3 | : | 23 | 37 | 1 |   | D | : | 61 |   |
| 7  |   | A | : | 25 | 0 | 2 |   | 3 | : | 27 | 5  | 1 |   | D | : | 33 |   |
| 8  |   | A | : | 29 | 0 | 3 |   | 4 | : | 32 | 19 | 2 |   | D | : | 53 |   |
| 9  |   | A | : | 33 | 0 | 2 |   | 3 | : | 35 | 9  | 1 |   | D | : | 45 |   |
| 10 |   | A | : | 37 | 0 | 1 |   | 1 | : | 38 | 25 | 2 |   | D | : | 65 |   |

Table I.101: Heuristic solution of the 10M3.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |   |            |
|---------------------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|---|------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 65 | 2 | D : | 69 |   |            |
| 2                                                 | A : | 5  | 0 | 1 | 1 : | 6  | 37 | 2 | D : | 45 |   |            |
| 3                                                 | A : | 9  | 0 | 2 | 2 : | 11 | 43 | 3 | D : | 57 |   |            |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 17 | 2 | D : | 33 |   |            |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 27 | 3 | D : | 49 |   |            |
| 6                                                 | A : | 21 | 0 | 2 | 3 : | 23 | 37 | 1 | D : | 61 |   |            |
| 7                                                 | A : | 25 | 0 | 2 | 3 : | 27 | 9  | 1 | D : | 37 |   |            |
| 8                                                 | A : | 29 | 0 | 2 | 2 : | 31 | 14 | 1 | 4 : | 46 | 5 | 2   D : 53 |
| 9                                                 | A : | 33 | 0 | 3 | 4 : | 36 | 27 | 2 | D : | 65 |   |            |
| 10                                                | A : | 37 | 0 | 1 | 1 : | 38 | 1  | 2 | D : | 41 |   |            |

Table I.102: Heuristic solution of the 10M4.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |   |            |
|---------------------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|---|------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 57 | 2 | D : | 61 |   |            |
| 2                                                 | A : | 5  | 0 | 1 | 1 : | 6  | 45 | 2 | D : | 53 |   |            |
| 3                                                 | A : | 9  | 0 | 1 | 1 : | 10 | 36 | 2 | D : | 48 |   |            |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 23 | 2 | D : | 39 |   |            |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 7  | 3 | D : | 29 |   |            |
| 6                                                 | A : | 21 | 0 | 2 | 2 : | 23 | 0  | 1 | 4 : | 24 | 9 | 2   D : 35 |
| 7                                                 | A : | 25 | 4 | 2 | 2 : | 31 | 10 | 3 | D : | 44 |   |            |
| 8                                                 | A : | 29 | 2 | 2 | 2 : | 33 | 6  | 2 | 3 : | 41 | 7 | 1   D : 49 |
| 9                                                 | A : | 33 | 2 | 2 | 3 : | 37 | 19 | 1 | D : | 57 |   |            |
| 10                                                | A : | 37 | 7 | 2 | 2 : | 46 | 16 | 3 | D : | 65 |   |            |

Table I.103: Heuristic solution of the 10M5.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |   |            |
|---------------------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|---|------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 53 | 2 | D : | 57 |   |            |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 51 | 3 | D : | 61 |   |            |
| 3                                                 | A : | 9  | 0 | 1 | 1 : | 10 | 21 | 2 | D : | 33 |   |            |
| 4                                                 | A : | 13 | 0 | 2 | 2 : | 15 | 35 | 3 | D : | 53 |   |            |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 19 | 3 | D : | 41 |   |            |
| 6                                                 | A : | 21 | 0 | 1 | 1 : | 22 | 5  | 2 | D : | 29 |   |            |
| 7                                                 | A : | 25 | 0 | 2 | 2 : | 27 | 10 | 1 | 1 : | 38 | 9 | 2   D : 49 |
| 8                                                 | A : | 29 | 0 | 2 | 3 : | 31 | 5  | 1 | D : | 37 |   |            |
| 9                                                 | A : | 33 | 0 | 3 | 4 : | 36 | 27 | 2 | D : | 65 |   |            |
| 10                                                | A : | 37 | 4 | 1 | 1 : | 42 | 1  | 2 | D : | 45 |   |            |

Table I.104: Heuristic solution of the 10M6.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 49 | 2 | D : | 53 |  |  |
| 2                                     | A : | 5  | 0 | 2 | 2 : | 7  | 55 | 3 | D : | 65 |  |  |
| 3                                     | A : | 9  | 0 | 2 | 2 : | 11 | 43 | 3 | D : | 57 |  |  |
| 4                                     | A : | 13 | 0 | 1 | 1 : | 14 | 17 | 2 | D : | 33 |  |  |
| 5                                     | A : | 17 | 0 | 2 | 2 : | 19 | 27 | 3 | D : | 49 |  |  |
| 6                                     | A : | 21 | 0 | 2 | 2 : | 23 | 13 | 3 | D : | 39 |  |  |
| 7                                     | A : | 25 | 0 | 1 | 1 : | 26 | 1  | 2 | D : | 29 |  |  |
| 8                                     | A : | 29 | 0 | 2 | 3 : | 31 | 9  | 1 | D : | 41 |  |  |
| 9                                     | A : | 33 | 0 | 3 | 4 : | 36 | 23 | 2 | D : | 61 |  |  |
| 10                                    | A : | 37 | 2 | 1 | 1 : | 40 | 3  | 2 | D : | 45 |  |  |

Table I.105: Heuristic solution of the 10M7.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |   |            |
|---------------------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|---|------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 38 | 2 | D : | 42 |   |            |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 47 | 3 | D : | 57 |   |            |
| 3                                                 | A : | 9  | 0 | 2 | 3 : | 11 | 57 | 1 | D : | 69 |   |            |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 21 | 2 | D : | 37 |   |            |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 23 | 3 | D : | 45 |   |            |
| 6                                                 | A : | 21 | 0 | 2 | 3 : | 23 | 29 | 1 | D : | 53 |   |            |
| 7                                                 | A : | 25 | 0 | 3 | 4 : | 28 | 31 | 2 | D : | 61 |   |            |
| 8                                                 | A : | 29 | 0 | 1 | 1 : | 30 | 1  | 2 | D : | 33 |   |            |
| 9                                                 | A : | 33 | 0 | 2 | 3 : | 35 | 13 | 1 | D : | 49 |   |            |
| 10                                                | A : | 37 | 0 | 3 | 4 : | 40 | 17 | 1 | 3 : | 58 | 6 | 1   D : 65 |

Table I.106: Heuristic solution of the 10M8.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|--|
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 45 | 2 | D : | 49 |  |  |
| 2                                     | A : | 5  | 0 | 2 | 2 : | 7  | 55 | 3 | D : | 65 |  |  |
| 3                                     | A : | 9  | 0 | 2 | 3 : | 11 | 61 | 1 | D : | 73 |  |  |
| 4                                     | A : | 13 | 0 | 1 | 1 : | 14 | 26 | 2 | D : | 42 |  |  |
| 5                                     | A : | 17 | 0 | 2 | 2 : | 19 | 39 | 3 | D : | 61 |  |  |
| 6                                     | A : | 21 | 0 | 2 | 3 : | 23 | 21 | 1 | D : | 45 |  |  |
| 7                                     | A : | 25 | 0 | 3 | 4 : | 28 | 39 | 2 | D : | 69 |  |  |
| 8                                     | A : | 29 | 0 | 2 | 2 : | 31 | 19 | 3 | D : | 53 |  |  |
| 9                                     | A : | 33 | 0 | 1 | 1 : | 34 | 1  | 2 | D : | 37 |  |  |
| 10                                    | A : | 37 | 0 | 3 | 4 : | 40 | 15 | 2 | D : | 57 |  |  |

Table I.107: Heuristic solution of the 10M9.t instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |    |   |     |    |  |  |
|---------------------------------------|-----|----|---|---|-----|----|----|---|-----|----|--|--|
| -----                                 |     |    |   |   |     |    |    |   |     |    |  |  |
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 66 | 2 | D : | 70 |  |  |
| 2                                     | A : | 5  | 0 | 1 | 1 : | 6  | 42 | 2 | D : | 50 |  |  |
| 3                                     | A : | 9  | 0 | 2 | 2 : | 11 | 60 | 3 | D : | 74 |  |  |
| 4                                     | A : | 13 | 0 | 2 | 2 : | 15 | 36 | 3 | D : | 54 |  |  |
| 5                                     | A : | 17 | 0 | 2 | 3 : | 19 | 46 | 1 | D : | 66 |  |  |
| 6                                     | A : | 21 | 0 | 1 | 1 : | 22 | 22 | 2 | D : | 46 |  |  |
| 7                                     | A : | 25 | 0 | 3 | 4 : | 28 | 48 | 2 | D : | 78 |  |  |
| 8                                     | A : | 29 | 0 | 2 | 3 : | 31 | 26 | 1 | D : | 58 |  |  |
| 9                                     | A : | 33 | 0 | 3 | 4 : | 36 | 24 | 2 | D : | 62 |  |  |
| 10                                    | A : | 37 | 0 | 1 | 1 : | 38 | 2  | 2 | D : | 42 |  |  |

Table I.108: Heuristic solution of the 10M10.t instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |             |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|-------------|
| -----                                             |     |    |   |   |     |    |     |   |     |     |    |             |
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 83  | 2 | D : | 87  |    |             |
| 2                                                 | A : | 5  | 0 | 1 | 1 : | 6  | 40  | 1 | 3 : | 47  | 51 | 1   D : 99  |
| 3                                                 | A : | 9  | 0 | 1 | 1 : | 10 | 34  | 1 | 3 : | 45  | 65 | 1   D : 111 |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 28  | 1 | 3 : | 43  | 93 | 1   D : 137 |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 113 | 3 | D : | 135 |    |             |
| 6                                                 | A : | 21 | 0 | 2 | 2 : | 23 | 95  | 1 | 4 : | 119 | 26 | 2   D : 147 |
| 7                                                 | A : | 25 | 0 | 2 | 2 : | 27 | 89  | 1 | 4 : | 117 | 40 | 2   D : 159 |
| 8                                                 | A : | 29 | 0 | 2 | 2 : | 31 | 83  | 1 | 4 : | 115 | 54 | 2   D : 171 |
| 9                                                 | A : | 33 | 0 | 2 | 3 : | 35 | 147 | 1 | D : | 183 |    |             |
| 10                                                | A : | 37 | 0 | 3 | 4 : | 40 | 153 | 2 | D : | 195 |    |             |

Table I.109: Heuristic solution of the 10S.s instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |  |  |
|---------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|--|--|
| -----                                 |     |    |   |   |     |    |     |   |     |     |  |  |
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 230 | 2 | D : | 234 |  |  |
| 2                                     | A : | 5  | 0 | 1 | 1 : | 6  | 214 | 2 | D : | 222 |  |  |
| 3                                     | A : | 9  | 0 | 1 | 1 : | 10 | 198 | 2 | D : | 210 |  |  |
| 4                                     | A : | 13 | 0 | 1 | 1 : | 14 | 182 | 2 | D : | 198 |  |  |
| 5                                     | A : | 17 | 0 | 2 | 2 : | 19 | 164 | 3 | D : | 186 |  |  |
| 6                                     | A : | 21 | 0 | 2 | 2 : | 23 | 148 | 3 | D : | 174 |  |  |
| 7                                     | A : | 25 | 0 | 2 | 2 : | 27 | 132 | 3 | D : | 162 |  |  |
| 8                                     | A : | 29 | 0 | 2 | 2 : | 31 | 116 | 3 | D : | 150 |  |  |
| 9                                     | A : | 33 | 0 | 2 | 3 : | 35 | 102 | 1 | D : | 138 |  |  |
| 10                                    | A : | 37 | 0 | 2 | 3 : | 39 | 86  | 1 | D : | 126 |  |  |

Table I.110: Heuristic solution of the 10C.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |   |         |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|---|---------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 83  | 2 | D : | 87  |    |   |         |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 125 | 3 | D : | 135 |    |   |         |
| 3                                                 | A : | 9  | 0 | 2 | 3 : | 11 | 159 | 1 | D : | 171 |    |   |         |
| 4                                                 | A : | 13 | 0 | 2 | 2 : | 15 | 81  | 3 | D : | 99  |    |   |         |
| 5                                                 | A : | 17 | 0 | 2 | 3 : | 19 | 127 | 1 | D : | 147 |    |   |         |
| 6                                                 | A : | 21 | 0 | 3 | 4 : | 24 | 157 | 2 | D : | 183 |    |   |         |
| 7                                                 | A : | 25 | 2 | 2 | 3 : | 29 | 81  | 1 | D : | 111 |    |   |         |
| 8                                                 | A : | 29 | 2 | 3 | 4 : | 34 | 123 | 2 | D : | 159 |    |   |         |
| 9                                                 | A : | 33 | 4 | 3 | 4 : | 40 | 109 | 1 | 2 : | 150 | 42 | 3 | D : 195 |
| 10                                                | A : | 37 | 6 | 3 | 4 : | 46 | 75  | 2 | D : | 123 |    |   |         |

Table I.111: Heuristic solution of the 10M1.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |   |                        |
|---------------------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|---|------------------------|
| 1                                                             | A : | 1  | 0 | 1 | 1 : | 2  | 143 | 2 | D : | 147 |    |   |                        |
| 2                                                             | A : | 5  | 0 | 1 | 1 : | 6  | 79  | 2 | D : | 87  |    |   |                        |
| 3                                                             | A : | 9  | 0 | 2 | 2 : | 11 | 145 | 3 | D : | 159 |    |   |                        |
| 4                                                             | A : | 13 | 0 | 2 | 2 : | 15 | 81  | 3 | D : | 99  |    |   |                        |
| 5                                                             | A : | 17 | 0 | 2 | 3 : | 19 | 151 | 1 | D : | 171 |    |   |                        |
| 6                                                             | A : | 21 | 0 | 2 | 3 : | 23 | 87  | 1 | D : | 111 |    |   |                        |
| 7                                                             | A : | 25 | 0 | 3 | 4 : | 28 | 153 | 2 | D : | 183 |    |   |                        |
| 8                                                             | A : | 29 | 2 | 3 | 4 : | 34 | 87  | 2 | D : | 123 |    |   |                        |
| 9                                                             | A : | 33 | 4 | 3 | 4 : | 40 | 75  | 1 | 3 : | 116 | 44 | 1 | 1 : 161 32 2   D : 195 |
| 10                                                            | A : | 37 | 6 | 3 | 4 : | 46 | 67  | 1 | 2 : | 114 | 18 | 3 | D : 135                |

Table I.112: Heuristic solution of the 10M2.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |   |         |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|---|---------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 107 | 2 | D : | 111 |    |   |         |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 161 | 3 | D : | 171 |    |   |         |
| 3                                                 | A : | 9  | 0 | 1 | 1 : | 10 | 75  | 2 | D : | 87  |    |   |         |
| 4                                                 | A : | 13 | 0 | 2 | 2 : | 15 | 129 | 3 | D : | 147 |    |   |         |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 101 | 3 | D : | 123 |    |   |         |
| 6                                                 | A : | 21 | 0 | 2 | 3 : | 23 | 159 | 1 | D : | 183 |    |   |         |
| 7                                                 | A : | 25 | 0 | 2 | 3 : | 27 | 71  | 1 | D : | 99  |    |   |         |
| 8                                                 | A : | 29 | 0 | 3 | 4 : | 32 | 125 | 2 | D : | 159 |    |   |         |
| 9                                                 | A : | 33 | 2 | 3 | 4 : | 38 | 95  | 2 | D : | 135 |    |   |         |
| 10                                                | A : | 37 | 4 | 3 | 4 : | 44 | 81  | 2 | 1 : | 127 | 66 | 2 | D : 195 |

Table I.113: Heuristic solution of the 10M3.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |             |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|-------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 203 | 2 | D : | 207 |    |             |
| 2                                                 | A : | 5  | 0 | 1 | 1 : | 6  | 127 | 2 | D : | 135 |    |             |
| 3                                                 | A : | 9  | 0 | 2 | 2 : | 11 | 157 | 3 | D : | 171 |    |             |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 83  | 2 | D : | 99  |    |             |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 125 | 3 | D : | 147 |    |             |
| 6                                                 | A : | 21 | 0 | 2 | 3 : | 23 | 159 | 1 | D : | 183 |    |             |
| 7                                                 | A : | 25 | 0 | 2 | 3 : | 27 | 83  | 1 | D : | 111 |    |             |
| 8                                                 | A : | 29 | 0 | 2 | 2 : | 31 | 107 | 1 | 4 : | 139 | 18 | 2   D : 159 |
| 9                                                 | A : | 33 | 0 | 3 | 4 : | 36 | 157 | 2 | D : | 195 |    |             |
| 10                                                | A : | 37 | 2 | 3 | 4 : | 42 | 79  | 2 | D : | 123 |    |             |

Table I.114: Heuristic solution of the 10M4.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |     |             |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|-----|-------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 179 | 2 | D : | 183 |     |             |
| 2                                                 | A : | 5  | 0 | 1 | 1 : | 6  | 151 | 2 | D : | 159 |     |             |
| 3                                                 | A : | 9  | 0 | 1 | 1 : | 10 | 123 | 2 | D : | 135 |     |             |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 95  | 2 | D : | 111 |     |             |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 65  | 3 | D : | 87  |     |             |
| 6                                                 | A : | 21 | 0 | 2 | 2 : | 23 | 22  | 1 | 4 : | 46  | 51  | 2   D : 99  |
| 7                                                 | A : | 25 | 0 | 2 | 2 : | 27 | 16  | 1 | 4 : | 44  | 77  | 2   D : 123 |
| 8                                                 | A : | 29 | 0 | 2 | 2 : | 31 | 10  | 1 | 4 : | 42  | 103 | 2   D : 147 |
| 9                                                 | A : | 33 | 0 | 2 | 3 : | 35 | 135 | 1 | D : | 171 |     |             |
| 10                                                | A : | 37 | 0 | 3 | 4 : | 40 | 153 | 2 | D : | 195 |     |             |

Table I.115: Heuristic solution of the 10M5.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |             |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|-------------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 167 | 2 | D : | 171 |    |             |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 173 | 3 | D : | 183 |    |             |
| 3                                                 | A : | 9  | 0 | 1 | 1 : | 10 | 87  | 2 | D : | 99  |    |             |
| 4                                                 | A : | 13 | 0 | 2 | 2 : | 15 | 141 | 3 | D : | 159 |    |             |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 101 | 3 | D : | 123 |    |             |
| 6                                                 | A : | 21 | 0 | 1 | 1 : | 22 | 63  | 2 | D : | 87  |    |             |
| 7                                                 | A : | 25 | 0 | 2 | 2 : | 27 | 87  | 1 | 1 : | 115 | 30 | 2   D : 147 |
| 8                                                 | A : | 29 | 0 | 1 | 1 : | 30 | 10  | 1 | 3 : | 41  | 69 | 1   D : 111 |
| 9                                                 | A : | 33 | 0 | 2 | 3 : | 35 | 159 | 1 | D : | 195 |    |             |
| 10                                                | A : | 37 | 0 | 2 | 3 : | 39 | 95  | 1 | D : | 135 |    |             |

Table I.116: Heuristic solution of the 10M6.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |   |         |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|---|---------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 155 | 2 | D : | 159 |    |   |         |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 185 | 3 | D : | 195 |    |   |         |
| 3                                                 | A : | 9  | 0 | 2 | 2 : | 11 | 157 | 3 | D : | 171 |    |   |         |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 83  | 2 | D : | 99  |    |   |         |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 125 | 3 | D : | 147 |    |   |         |
| 6                                                 | A : | 21 | 0 | 2 | 2 : | 23 | 85  | 3 | D : | 111 |    |   |         |
| 7                                                 | A : | 25 | 0 | 1 | 1 : | 26 | 59  | 2 | D : | 87  |    |   |         |
| 8                                                 | A : | 29 | 0 | 1 | 1 : | 30 | 10  | 1 | 3 : | 41  | 81 | 1 | D : 123 |
| 9                                                 | A : | 33 | 0 | 2 | 3 : | 35 | 147 | 1 | D : | 183 |    |   |         |
| 10                                                | A : | 37 | 0 | 2 | 3 : | 39 | 95  | 1 | D : | 135 |    |   |         |

Table I.117: Heuristic solution of the 10M7.s instance.

| c   i : p s m   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |    |   |         |
|---------------------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|----|---|---------|
| 1                                                 | A : | 1  | 0 | 1 | 1 : | 2  | 119 | 2 | D : | 123 |    |   |         |
| 2                                                 | A : | 5  | 0 | 2 | 2 : | 7  | 161 | 3 | D : | 171 |    |   |         |
| 3                                                 | A : | 9  | 0 | 2 | 3 : | 11 | 195 | 1 | D : | 207 |    |   |         |
| 4                                                 | A : | 13 | 0 | 1 | 1 : | 14 | 95  | 2 | D : | 111 |    |   |         |
| 5                                                 | A : | 17 | 0 | 2 | 2 : | 19 | 113 | 3 | D : | 135 |    |   |         |
| 6                                                 | A : | 21 | 0 | 2 | 3 : | 23 | 135 | 1 | D : | 159 |    |   |         |
| 7                                                 | A : | 25 | 0 | 3 | 4 : | 28 | 153 | 2 | D : | 183 |    |   |         |
| 8                                                 | A : | 29 | 2 | 1 | 1 : | 32 | 65  | 2 | D : | 99  |    |   |         |
| 9                                                 | A : | 33 | 0 | 2 | 3 : | 35 | 111 | 1 | D : | 147 |    |   |         |
| 10                                                | A : | 37 | 0 | 3 | 4 : | 40 | 133 | 1 | 3 : | 174 | 20 | 1 | D : 195 |

Table I.118: Heuristic solution of the 10M8.s instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |  |  |  |
|---------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|--|--|--|
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 143 | 2 | D : | 147 |  |  |  |
| 2                                     | A : | 5  | 0 | 2 | 2 : | 7  | 185 | 3 | D : | 195 |  |  |  |
| 3                                     | A : | 9  | 0 | 2 | 3 : | 11 | 207 | 1 | D : | 219 |  |  |  |
| 4                                     | A : | 13 | 0 | 1 | 1 : | 14 | 107 | 2 | D : | 123 |  |  |  |
| 5                                     | A : | 17 | 0 | 2 | 2 : | 19 | 161 | 3 | D : | 183 |  |  |  |
| 6                                     | A : | 21 | 0 | 2 | 3 : | 23 | 111 | 1 | D : | 135 |  |  |  |
| 7                                     | A : | 25 | 0 | 3 | 4 : | 28 | 177 | 2 | D : | 207 |  |  |  |
| 8                                     | A : | 29 | 2 | 2 | 2 : | 33 | 123 | 3 | D : | 159 |  |  |  |
| 9                                     | A : | 33 | 2 | 1 | 1 : | 36 | 73  | 2 | D : | 111 |  |  |  |
| 10                                    | A : | 37 | 0 | 3 | 4 : | 40 | 129 | 2 | D : | 171 |  |  |  |

Table I.119: Heuristic solution of the 10M9.s instance.

| c   i : p s m   i : p s m   i : p s m |     |    |   |   |     |    |     |   |     |     |  |
|---------------------------------------|-----|----|---|---|-----|----|-----|---|-----|-----|--|
| 1                                     | A : | 1  | 0 | 1 | 1 : | 2  | 206 | 2 | D : | 210 |  |
| 2                                     | A : | 5  | 0 | 1 | 1 : | 6  | 142 | 2 | D : | 150 |  |
| 3                                     | A : | 9  | 0 | 2 | 2 : | 11 | 208 | 3 | D : | 222 |  |
| 4                                     | A : | 13 | 0 | 2 | 2 : | 15 | 144 | 3 | D : | 162 |  |
| 5                                     | A : | 17 | 0 | 2 | 3 : | 19 | 178 | 1 | D : | 198 |  |
| 6                                     | A : | 21 | 0 | 1 | 1 : | 22 | 114 | 2 | D : | 138 |  |
| 7                                     | A : | 25 | 0 | 3 | 4 : | 28 | 204 | 2 | D : | 234 |  |
| 8                                     | A : | 29 | 2 | 2 | 3 : | 33 | 140 | 1 | D : | 174 |  |
| 9                                     | A : | 33 | 2 | 3 | 4 : | 38 | 146 | 2 | D : | 186 |  |
| 10                                    | A : | 37 | 4 | 1 | 1 : | 42 | 82  | 2 | D : | 126 |  |

Table I.120: Heuristic solution of the 10M10.s instance.

### I.3 Large-scale class instances

Heuristic solutions of all large-scale class instances can be found on the enclosed CD-rom.